Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

Пояснювальна записка

до магістерської кваліфікаційної роботи
магістр
(освітньо-кваліфікаційний рівень)

на тему: Програмний засіб для приховування текстової інформації у цифровому зображенні

Виконав: студент 5 курсу, групи 2КІ-18м
спеціальності
123 – «Комп'ютерна інженерія»
(шифр і назва напряму підготовки, спеціальності)

Казаков Р.Г.
(прізвище та ініціали)
Керівник доц. каф. ОТ, к.т.н. Савицька Л.А.
(прізвище та ініціали)
Рецензент к.т.н., доц каф. МБІС Карпінець В.В.
(прізвище та ініціали)

м. Вінниця - 2019 рік

РЕФЕРАТ

Магістерська кваліфікаційна робота присвячена розробці програмного засобу для приховування текстової інформації у цифровому зображенні. Розроблено інтерфейс програми і запропонована комбінація алгоритмів, яка є більш стійкою до методів виявлення та надає додатковий захист у вигляді шифрування.

Розробка виконана в середовищі VScode, також було використано мову гіпертекстової розмітки HTML, SCSS, JavaScript та PHP.

У роботі також виконані економічні розрахунки доцільності створення нового програмного продукту.

REVIEW

The master's qualification is dedicated to the development of software for hiding text information in a digital image. A program interface has been developed and a combination of algorithms has been proposed that is more robust to detection methods and provides additional security in the form of encryption.

The development was done in VScode, HTML, SCSS, JavaScript, and PHP hypertext markup language was also used.

The economic calculations of the feasibility of creating a new software product were also made.

ЗМІСТ

ВСТУП 8
1.1 Становлення стеганографії як науки 11
1.2 Поняття стеганографії та її особливості 15
1.3 Типи контейнерів 18
1.4 Постановка задачі по приховуванню текстової інформації 19
1.5 Висновок до розділу 21
2 МЕТОДИ ТА АЛГОРИТМИ ДЛЯ СТЕГАНОГРАФІЧНОГО ПРИХОВУВАННЯ 22
2.1 Порівняння методів та програних засобів для приховування інформації 22
2.2 Методика і алгоритми приховування тексту в зображенні 27
2.3 Розробка алгоритму приховування текстової інформації 33
2.4 Висновок до розділу 36
3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ПРИХОВУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ 37
3.1 Розробка архітектури програмного продукту 37
3.2 Розробка програми приховування інформації 38
3.3 Перевірка працездатності програми 45
3.4 Висновок до розділу 52
4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ПРИХОВУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ У ЦИФРОВОМУЗОБРАЖЕННІ 53
4.1 Загальні положення 53
4.2 Оцінювання комерційного потенціалу розробки 54
4.3 Прогнозування витрат на виконання та впровадження результатів наукової роботи 59
4.4 Прогнозування комерційних ефектів від реалізації результатів розробки 64
4.5 Розрахунок ефективності вкладених інвестицій та періоду їх окупності 66
4.6 Висновок до розділу 69
ВИСНОВКИ 71
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 73
Додаток А 75
Додаток Б 78
Додаток В 79
Додаток Г 80
Додаток Д 81
Додаток Е 82
Додаток Ж 83
Додаток К 84
Додаток Л 85
Додаток М 86
Додаток Н 87

ВСТУП

Актуальність дослідження:Розвиток електронних комунікацій та Інтернет-сервісів відбувається в умовах постійного зростання кількості загрози їхній безпеці. Повідомлення про злом великих IT-систем і крадіжці даних публікуються в електронних ЗМІ регулярно. Оскільки сьогодні несанкціонований доступ до конфіденційної інформації став однією зі складових міжнародного кримінального бізнесу, захист інформації повинний бути постійно виконуваної функцією, інтегрованою з бізнес-процесами організації, а відповідні апаратно-програмні засоби потребують удосконалення.

Також, протягом останніх 15-20 років створено та розробляється безліч методів і алгоритмів стегано-криптографічного захисту інформації, а також відповідного програмного забезпечення для персональних комп'ютерів.

У той же час інтерес користувачів до програм стеганографічної захисту інформації істотно знизився, що пов'язано, в тому числі і з певним прогресом в області стеганоаналіза.

Аналіз ринку продуктів для захисту інформації показує, що за останні багато років не з'явилося жодного нового успішного інноваційного комерційного програмного продукту стегно-криптографічного захисту.

Відомі фірми – виробники або поповнюють стеганографічні програми додатковими інструментами комп'ютерної безпеки, або освоюють суміжні області інформаційних технологій.

Більшість стеганографічних програмних продуктів доступних в Інтернеті використовують спосіб найменш значущих біт (НСБ) для приховування інформації в контейнері[1].

У той же час відомо кілька методів стеганоаналіза, побудованих на основі дуальної статистики, які дозволяють не тільки визначити наявність в зображенні даних, прихованих за допомогою НСБ-алгоритмів, але і дати оцінку обсягу прихованої в контейнері інформації[2].

Тому прості стеганографічні методи не забезпечують таємницю передачі даних, тобто за ступенем захисту інформації не перевершують криптографічні продукти, що і пояснює падіння інтересу до них.

Рішення проблеми вимагає створення нових апаратно-програмних засобів стегано-криптографічного захисту інформації.

Приховування інформації (або стеганографія) направлена на безпосереднє приховування одних секретних (важливих) повідомлень в інші, причому ховається навіть саме існування секрету [3]. Як правило, відправник пише повідомлення яке на мові стеганографії називається контейнер. Потім виконується приховування конфіденційної інформації на цьому ж носії [4].

Даний напрямок захисту інформації дуже актуально в даний час, про що свідчать ресурси, що інвестуються в розвиток стеганографії урядами провідних країн світу та терористичними організаціями.

Метою дослідження є вдосконалення методу стеганографічного приховування текстової інформації в цифровому зображенні.

Задачі дсолідження:

– аналіз існуючих стеганографічних методів на основі технології найменш значущий біт (НЗБ) та окреслення їх недоліків;

– провести порівняльний аналіз переваг та недоліків в існуючих методах приховування інформації;

– навести методику приховування інформації у зображеннях, яка описувала порядок приховування довільного тексту в цифровому зображенні;

– розробити програму, яка передбачає взаємодію з віддаленим сервером на якому відбуватиметься приховування інформації;

– описати архітектуру роботи програми;

– перевірити працездатність програми;

– здійснити обгрунтування доцільності виконання нового наукового рішення та розробки програмного засобу.

Об'єкт дослідження – процес оброблення візуальної інформації дляприховування текстових даних.

Предмет дослідження – методи стеганографічного приховування текстових даних у зображеннях.

Методи дослідження:методи теорії чисел, методи дискретної математики, методи теорії інформації.

Наукова новизна одержаних результатів – удоскноналено метод стеганографічного захисту інформації, який відрізняється від існуючих поєднанням стеганографічного алгоритму LSB та криптографічного RSA, що дозволяє більш ефективно здійснювати процес приховування інформації.

Практичне значення одержаних результатів:

- створено алгоритм для приховування тектсової інформації у цифровому зображенні.
- розроблено програму для приховування текстової інформації у цифровому зображенні.

Апробація–зроблено доповідь на всеукраїнський науково-практичний Інтернет-конференції студентів, аспірантв та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи-2020».

Публікації [5]: Казаков Р. Г Алгоритм для приховування текстової інформації у цифровому зображенні./ Тези доповіді. Матеріали всеукраїнської науково-практичної Інтернет-конференції студентів, аспірантв та молодих науковців «Молодь в науці: дослідження, проблеми, перспективи-2020».

1 АНАЛІЗ ПІДХОДІВ СТЕГАНОГРАФІЧНОГО ЗАХИСТУ ІНФОРМАЦІЇ

1.1 Становленя стеганографії як науки

Стеганографічні методи використовували з давна та у різні часи але попри те, саме Єгипет називають місцем зародження цього напрямку, хоча наскальні рисунки по типу символьної техніки, також можна віднести до повідомлень з використанням стеганографії.

Провідним спогадом, який зіставляють з 440 р. до н. е., були два метода. Перший полягав у написанні повідомлення на дерев'яній підкладці до нанесення воскового шару. У другому способі повідомлення наносили на оголений череп раба, і коли волосся відростало – відправляли до адресата [3].

У Китаї повідомлення записували на тоненьких смужках шовку, які згортали та покривали воском. Він в свою чергу надавав їм кулько-подібної форми. Ці кульки посильні ковтали і таким чином транспортували необхідну інформацію.

В 1499 році роботи пов'язані з методами передачі прихованих повідомлень на основі криптографії і стеганографії XV-го століття, були об'єднані в книгу "Steganographia" [4].

XVII-XVIII століття видатні штатами "чорних кабінетів" які складались не тільки з криптографів та дешифрувальників, а також з хіміків. Їх присутність була обумовлена активним на той час використанням "чорнил невидимок". Крім самих чорнил, для передачі таємного повідомлення, використовували молоко як альтернативу. Однією із згадок про таке використання є повість "Біля витоків майбутнього".

В 2000 році американською школяркою був запропонований алгоритм, за допомогою якого можливо приховати повідомлення у генну послідовність ДНК[5].

Призначення комп'ютерної безпеки полягає в захисті інформації від несанкціонованого доступу, навмисної зміни даних з мінімальними втратами, які не помітні звичайному користувачеві і не впливають на основний функціонал файлів. Для таких операцій існують поняття криптографії та стеганографії. В той час як криптографія була створена з метою захисту каналів зв'язку завдяки засобам кодування та розшифровки даних, то

стеганографія націлена на приховання самого факту наявності такого каналу. Тому, актуальність проблеми полягає в тому, щоб обрати найбільш ефективний підхід для передачі важливої інформації. В поточній роботі використовується підхід вбудовування інформації в цифрові зображення.

Станом на сьогоднішній день існує безліч методів приховування повідомлень для конкретних видів контейнерів.

Рисунок 1.1– Типи файлів для використання в якості контейнера.

Перевагу для схову даних надають саме зображенням, так як в них можна здійснити приховання без спотворень і тому що в наш час доставка здійснюється через інтернет.

Будь-яке розширення зображень має послідовність зайвих бітів, користуючись якими втрати мінімальні, а якість залишається не змінною. В

процесі ущільнення зображення беруть до уваги розширення контейнера, це впливає на складність і обсяг файлу який можна вмонтувати.

Для вибору стеганографічного методу, існують певні критерії. Їх наявність обумовлює відповідність конкретній задачі, через те що всякий метод має переваги і недоліки залежно від ситуації.

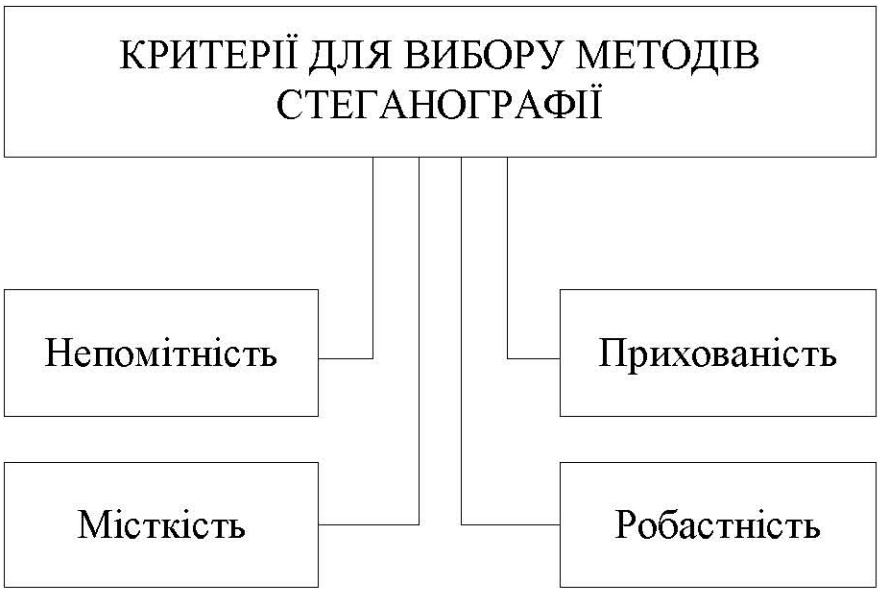

Рисунок 1.2 – Критерії методів приховування.

Непомітність – різниця між початковим та зображенням з прихованим повідомленням не повинна бути помітною для людського ока. Цей критерій є основним при виборі даного методу. Але слід зазначити, ця вимога є залежною від таких факторів як формат зображення та обсяг даних.

Прихованість – визначає успішність методу приховування, при розпізнанні наявності повідомлення, що обумовлює складність алгоритму розпізнавання.

Місткість – також залежить від формату зображення і являє собою оцінку розміру даних.

Робастність – полягає в відсутності втрат даних в процесі обробки або передачі. Мова йдеться про втрати, які властиві обраному контейнеру.

Актуальною проблемою є те, що алгоритм має відповідати всім критеріям, а тому вибір алгоритму є не менш важливим фактором. Розглянемо існуючі алгоритми для різних контейнерів.

LSB для растрових зображень BMP. Для приховування повідомлень в такі файли необхідно якнайбільший контейнер-зображення. До того ж на сьогоднішній день у мережі інтернет, зображення BMP формату використовують рідко, в першу чергу через відсутність стиснення, тому багаторазова передача даних у цьому форматі може викликати підозри. Ще однією причиною не відповідності критеріям методів приховування даного алгоритму є те, що повідомлення приховані в формат BMP можуть мати занадто великий розмір. А це в свою чергу змінює більшу кількість бітів і збільшує різницю вихідного зображення від початкового, що є грубим порушенням критерія прихованості.

LSB для зображень формату JPEG. Цей формат використовує 8 бітів на кожен колір RGB і всього 24 біта на піксель, тому може вмістити в собі повідомлення великого обсягу, залишаючись при цьому максимально незмінним, але це залежить від методу який застосовують. Недолік JPEG – наявність пошкоджень повідомлення. Так звані пошкодження не обов'язково будуть, але шанс їх появи великий через використання JPEG стиснення з втратами в процесі стиснення повідомлення. В цьому випадку порушеним критерієм є робастність.

Метод використання патчів – розкладає контейнер на менші фрагменти, що дозволяє вставити більшу кількість бітів. Повідомлення розподіляється рівномірно по всьому зображенню і навіть якщо в процесі обробки використовують стиснення з втратами, пошкодженні патчі можна відновити з інших. Але це залежить від розміру прихованої інформації. Якщо повідомлення не дуже довгі, його можна продублювати. Недоліком є інкапсуляція одного патчу в один біт.

Формат зображення для методу псевдовипадкової перестановки не має значення. Суть його зводиться до того, що біти інформації яку ми хочем приховати, вбудовуються з зміною порядку їх появи в повідомленні. Цей випадок аналогічний методу молодшого біта LSB, вбудованого випадковою послідовністю, тобто приховувана інформація може зазнати втрат. Якщо ж вбудовувати не в останні, а будь-які інші біти, то ймовірно можна буде

спостерігати появу зайвих шумів на зображенні. Порушення критеріїв прихованості та робастності.

1.2 Поняття стеганографії та її особливості

Стеганографія (з грецьк. steganos – секрет, таємниця; graphy – запис) – це приховування інформації в об'єктах різних форматів, при цьому повідомлення вбудоване в контейнер, не привертає уваги, звісно якщо не знати про факт існування такого повідомлення [6]. Це і є основна відмінність від криптографії. Але слід зазначити, що стеганографія ні в якому разі не заміна криптографії, скоріше доповнення. Якщо інформацію приховати методами стеганографії, то буде зменшена ймовірність вияву факту передачі інформації. Якщо ж додатково таку інформацію зашифрувати, то отримаєм додатковий рівень захисту.

При тому що стеганографія як метод приховування інформації відома здавна, комп'ютерна стеганографія – достатньо молоде відгалуження. І як всяке новоутворення, навіть при великій кількості відкритих конференцій та доступних публікацій, не мала загально-визначеної термінології довгий час. На конференції Information Hiding: First Information Workshop 1996 року К. Шеннон запропонував єдину модель стеганографічної системи, яка полягала в наступному: У сучасній комп'ютерній стеганографії існує два основних типи файлів. Перший – файл-повідомлення, який використовують для того щоб приховати. Другий – файл-контейнер, безпосередньо в нього і записують файл-повідомлення. Файли-контейнери розрізняють наступним чином. Якщо в контейнер ще не записана інформація, він є порожнім, і має назву контейнер-оригінал. Якщо ж в контейнері вже є вбудована інформація, він заповнений, і називається контейнер-результат. Коли використовують назву "ключ" мова йде про секретний елемент, функція якого передбачати порядок запису файл-повідомлення в контейнер-оригінал.

Основні положення сучасної комп'ютерної стеганографії:

- методи приховання повинні забезпечувати автентичність і цілісність файлу;
- передбачається, що зловмиснику відомі всі можливі стеганографічні методи;

- безпека методів ґрунтується на збереженні стеганографічних перетворень основних властивостей переданого файлу при внесені до нього секретного повідомлення і деякої невідомої зловмиснику інформації – ключа;
- навіть якщо факт приховання став відомий зловмиснику, отримання секретного повідомлення є складною обчислювальною задачею.

Станом на сьогоднішній день стеганографічними методами користуються для вирішення задач:

- захист конфіденційної інформації від несанкціонованого доступ;
- подолання систем моніторингу та управління мережевими ресурсами;
- маскування ПЗ;
- захист авторського права на окремі види інтелектуальної власності.

Стегосистема – є об'єднанням методів та засобів, які спрямовані на створення прихованого каналу зв'язку, та мають відповідати чітко визначеному переліку принципів для коректної побудови.

Перелік принципів стеганографічної системи:

- зловмисник повністю проінформований відносно стегосистеми та подробиць її створення. Єдина інформація, яка невідома – ключ. Він необхідний, щоб визначити присутність прихованого повідомлення і як наслідок його вміст.
- якщо зловмисник вже знає про факт приховання повідомлення, саме зберігання цілісності ключа, не дозволить отримати повідомлення з даних в які вони вбудовані.
- зловмисник має бути позбавлений можливості використовувати будь-які технічні засоби, тим більше мати перевагу в дешифруванні повідомлень.

Вбудоване повідомлення – інформація, призначенням якої є передавання по каналам зв'язку, шляхом вбудовування в контейнер. Такі повідомлення не обов'язково є текстовими, так само можуть передаватись зображення або звукові файли.

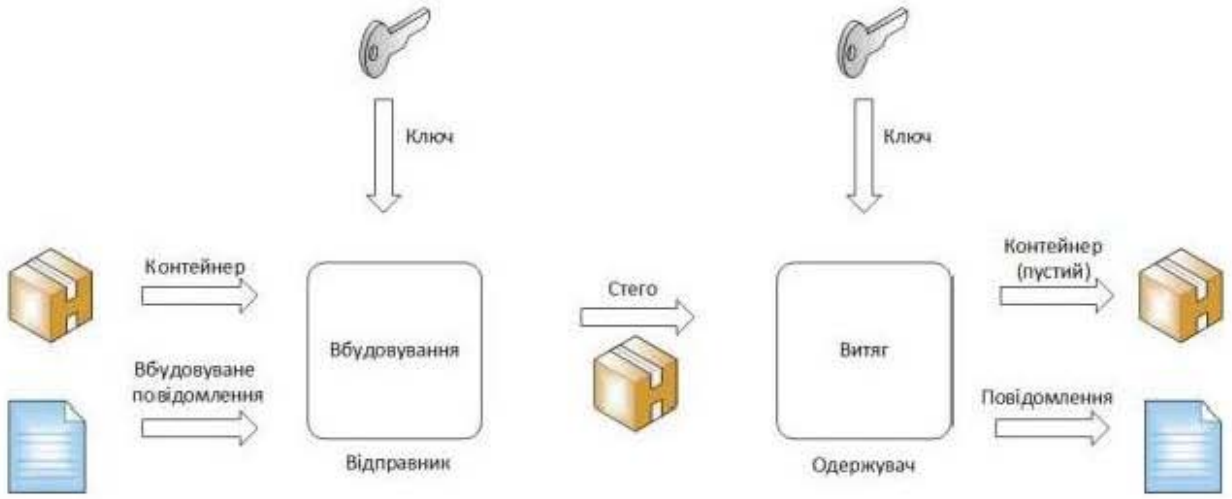

Рисунок 1.3 – Узагальнена модель стегосистеми.

Контейнер – інформація, в яку безпосередньо і буде приховуватись повідомлення.

Порожнім контейнером – називають такий контейнер, який не містить прихованої інформації.

Стеганографічний канал (стего) – канал зв'язку, по якому передають заповнений контейнер.

Ключ – секретний ключ, потрібний для вбудовування даних. Стеганографічна система не обумовлена наявністю всього лише одного ключа, їх може бути безліч. Це залежить від наявності та кількості рівнів захисту [7]. Існує два типи стегоключа – секретний та відкритий ключ. Стегосистема з секретним ключем являє собою застосування одного ключа, який відомий для обох сторін до початку обміну вбудованою інформацією. В той час як стегосистеми з відкритим ключем застосовують різні ключі для приховання та виявлення інформації. Завдяки чому вивести за допомогою обчислень один ключ з іншого не можливо. Це дає змогу передавати ключ по незахищеному каналу зв'язку.

Але слід пам'ятати, що при використанні будь-яких стеганографічних методів для приховування інформації, існує певна залежність між надійністю та розміром повідомлення. Чим більшу надійність нам необхідно отримати наприкінці процесу приховування, тим менший обсяг даних повинен бути

прихований в контейнері. Іншими словами, вибір контейнеру обумовлює розмір інформації, яку можливо приховати з високою надійністю.

1.3 Типи контейнерів

Вибір контейнера відіграє важливу роль у приховуванні інформації. Крім типів їх поділяють за протяжністю.

Потокові контейнери – в таких контейнерах значно важче визначити де він починається і де закінчується, неможливо завчасно визначити наступний шумовий біт, так як його визначає спеціальний генератор, який також визначає відстань між ними. Тому вбудовування інформації відбувається у масштабі реального часу [8]. Найбільш складним при отриманні повідомлень з безперервного потоку даних є визначення його початку, але в потоковому контейнері можуть бути сигнали синхронізації, так звані “кордони пакета”, саме вони будуть свідчити про початок прихованого повідомлення. На стороні відправника також можуть бути перешкоди, які проявляються у довжині потоку. Тобто відправник повинен бути впевнений, що довжини потоку вистачить для передачі прихованої інформації.

Фіксовані контейнери – передбачають, що сторона відправника чітко знає розмір повідомлення. До того ж використовуючи даний тип контейнера, є можливість обирати послідовність шумових бітів. Одним з недоліків такого контейнера є обмежена величина, це може призвести до того, що повідомлення просто не поміститься у контейнері. Але на практиці через поширеність і доступність використовують саме такі контейнери.

Різноманітність підходів не обмежується потоковими чи фіксованими контейнерами.

Існують контейнери, які генеруються самою стегосистемою. Одна з програм, яка використовує такий метод для вбудовування інформації, – генерує фрактал Мандельборта. Має назву даний підхід – конструююча стеганографія.

Селектуюча стеганографія – передбачає генерацію кількох можливих варіантів контейнера, серед яких потрібно вибрати потрібний або більш актуальний. І в обраний контейнер приховується інформація.

Безальтернативна стеганографія – окреслює надходження контейнера ззовні. Можливість вибору в такому підході відсутня, тому для приховування береться перший контейнер. Навіть якщо він не дуже підходить.

Якістю контейнера називається величина, що дорівнює відношенню обсягу вбудованої обраним стеганографічним алгоритмом інформації до розміру оцінюваного контейнера, що виражена у відсотках. Обсяг інформації, яку потрібно приховати визначається методом вбудовування, і обчислюється формулою:

$$C = \sum_{i=1}^{n} N, \qquad (1.1)$$

де N – число інтервалів в і-му рядку;

n – число непустих рядків в тексті.

Дана формула дає змогу обчислити максимальний обсяг безпечно вбудованої інформації.

1.4 Постановка задачі по приховуваню текстової інформації

В наш час, коли кількість використання цифрових ресурсів сягає незліченних обсягів, дослідження і впровадження стеганографії все більше схоже на вимушену міру [9]. Спілкування і обмін інформацією щоденно безлічі користувачів, що доволі часто є особистою, супроводжується незаконним копіюванням і розповсюдженням. Нерідко з метою збагачення. В наслідок таких дій, підвищується жага до приватності і захист особистої інформації.

Завдання стеганографії – передача повідомлення обумовлена прихованим змістом, і фактом його наявності. Для реалізації передачі повідомлень таким чином потрібно мати контейнер. Це данні в які вбудовуватиметься повідомлення. Такий контейнер не повинен привертати уваги, тому зміни після приховання повинні бути мінімальними, інакше порушення дозволеної кількості біт, які можна змінити, призведе до значних втрат якості контейнера. І обмін графічними файлами не буде вважатись звичайним. В будь-якому разі є правило, яким нехтувати не слід. Ніхто не

повинен мати в наявності початковий файл, який використовувався в якості контейнера, та кінцевий – з прихованим змістом. Тому що розкрити наявність прихованої інформації дозволить просте порівняння файлів. У переважній більшості використовують файли з розширенням BMP, так як вони вважаються найкращими у якості контейнера для стеганографічних маніпуляцій [10]. Причиною тому є те що в них висока якість зображення. До того ж вони дозволяють виконувати стиснення без втрат. Розширення BMP не єдине в якому використовуються подібні методи, доповнюють його такі розширення як TIFF, TGA, PNG і т.д.

До мети даної магістерської роботи відноситься:

- визначення актуальності проблеми;
- аналіз існуючих стеганографічних методів на основі технології найменш значущий біт (НЗБ) та окреслення їх недоліків;
- порівняння засобів для приховування інформації.
- опис вдосконаленого методу приховування текстових повідомлень у цифрових зображеннях.
- перевірка працездатності веб-інтерфейсу.
- практична реалізація інтерфейсу, який передбачає взаємодію з віддаленою програмою, робота якої базується на покращеному стеганографічному алгоритмі.

1.5 Висновок до розділу

У рамках першого розділу розглянуто основні етапи розвитку направлення стеганографії, проведений аналіз основних понять, компонентів і методів комп'ютерної стеганографії, які пов'язані з вбудовуванням тексту в зображення різних форматів. Для вирішення поставленої задачі створено поетапний та структурований перелік завдань.

2 МЕТОДИ ТА АЛГОРИТМИ ДЛЯ СТЕГАНОГРАФІЧНОГО ПРИХОВУВАННЯ

2.1 Порівняння методів та програних засобів для приховування інформації

Основними положеннями сучасної комп'ютерної стеганографії відносно методів прихування інформації є такі:

– методи приховування інформації повинні забезпечувати автентичність і цілісність файлу;

– передбачається, що противнику (зловмисникові) повністю відомі використання можливих стеганографічних методів;

– безпека методів повинна грунтуватись на збереженні основних властивостей відкрито переданого файлу-контейнера.

– навіть якщо факт прихування став відомий противнику через спільника, то витяг самого файлу-повідомлення повинно собою являти складну обчислювальну задачу.

В даний час методи комп'ютерної стеганографії розвиваються за двома основними напрямками:

– методи, що засновані на використанні спеціальних властивостей комп'ютерних форматів;

– методи, що засновані на використанні надмірності аудіо та візуальної інформації.

Наведемо таблицю порівняльний аналіз переваг та недоліків в існуючих методах прихування інформації (табл. 1.1).

Насамперед, методи використання зарезервованих для розширення полів комп'ютерних форматів даних полягає у тому, що в багатьох мультимедійних форматах існують поля розширення які заповнюються нульовою інформацією і не враховуються програмою.

Методи, що використовують відомі зміщенні слова, речення, абзаци базуються на зміні положення рядків і розстановки слів у реченні. Це забезпечується вставкою додаткових пробілів між словами.

Таблиця 1.1–Порівняльний аналіз переваг та недоліків в існуючих методах приховання інформації

Методи приховування інформації	Недоліки	Переваги
Методи, що використовують зарезервовані розширення полів.	Низький ступінь прихованості, можливість передавати файли обмеженого обсягу.	Легкість у використанні.
Методи, що використовують відомі зміщенні слова, речення, абзаци.	Мало продуктивний метод, можливість передавати файли обмеженого обсягу.	Легкість у використанні.
Методи вибору певних позицій букв	Низький ступінь прихованості	Існує велика кількість вільно поширеного програмного забезпечення для реалізації цих методів.
Методи приховування інформації в деяких місцях гнучких дисків	Мало продуктивний метод, можливість передавати файли обмеженого обсягу.	Простота використання.
Методи, що використовують надмірності файлів	Спотворення статистичних характеристик цифрових потоків.	Можливість прихованої передачі великого обсягу інформації.

Методи вибору певних позицій букв (нульовий шифр) являє собою «Акровірш». Окремий випадок цього методу, наприклад, початкові літери кожного рядка утворюють повідомлення.

Не досить розповсюдженими є методи що використовують спеціальні властивості полів форматів, які не відображаються на екрані та засновані на використанні спеціальних «невидимих», прихованих полів для організації виносок і посилань, наприклад, використання чорного шрифту на чорному фону.

Також не досить частими є методи приховування інформації в місцях гнучких дисків що не використовуються операційною та файловою системою, наприклад, в нульовий доріжці диску.

Методи, що використовують надмірності файлів цифрового відеоряду, фотографій або цифрового звуку містять молодші розряди цифрових відліків і мають дуже мало корисної інформації. Їх заповнення додатковою інформацією практично не впливає на якість сприйняття, що і забезпечує можливість приховування. До головних недоліків цього методу відноситься те, що зі зміною інформації спотворюються статистичні характеристики цифрових потоків. Для зниження компрометуючих ознак потрібна корекція статистичних характеристик.

Однак, цей метод досить розповсюджений внаслідок можливість приховати та передати великі обсяги інформації. Також, існує можливість додаткового захисту авторського права, прихованого зображення, товарної марки, реєстраційних номерів та інше.

Досвід комерціалізації і практичного використання програм найменш значущих біт (НЗБ) стеганографії з зображеннями або зі звуковими файлами показує, що ці програми–індивідуальні засоби приховування інформації.

За останні 20 років існування стеганографічних програм не було прикладу їх масового довгострокового впровадження в інформаційну мережу організацій, інтеграції з бізнес-процесами або побудови на їх основі діючих протоколів прихованої передачі даних.

Це можна пояснити наступними причинами.

Широкомасштабне розгортання засобів з приховування інформації в корпоративному середовищі суперечить суті їх застосування секретності використання, оскільки подібну діяльність складно довго зберегти в таємниці. Дійсно, якщо зловмиснику відомо, що для передачі даних впливають криптографічні програми, то ефект від їх використання для захисту інформації значно знижується.

Робота із засобами, що приховують інформацію передбачає, по крайній мірі, розуміння відмінностей між форматами графічних файлів, а також базового рівня знань в області захисту інформації для усунення загрози

втрати конфіденційних даних через помилки, пов'язаних з людським фактором.

Активне використання програм, що приховують інформацію вимагає захищеного зберігання великого масиву унікальних або маловідомих зображень або наявності у користувача навичок по вибору з мережі інтернет цифрових носіїв, що відповідають поставленим завданням.

Такий поширений метод приховування інформації, як НЗБ-стеганографія в порівнянні з криптографією, істотно обмежує пропускну здатність каналу зв'язку, оскільки в цілях таємності передачі даних не рекомендується заповнювати зображення-носій більш ніж на 10% ємності НЗБ-рівня.

Цим пояснюється відсутність комерційних протоколів прихованої передачі даних на базі НЗБ-стеганографії.

Використання стеганографічного програмного забезпечення при дотриманні заходів комп'ютерної безпеки має рівнозначну захищеність інформації з обмеженим доступом при роботі на корпоративних персональних комп'ютерах, ноутбуках або при запуску з зовнішнього накопичувача з апаратно-програмним шифруванням потоку даних. З іншого боку, флеш-накопичувачі, навіть в корпоративному середовищі – це персоналізовані пристрої зберігання інформації.

Тому, поєднання флеш-накопичувача і сервісного програмного забезпечення, в тому числі стеганографічного, видається природним з позиції підвищення функціональності і ефективності використання пристроїв зберігання даних і програмного забезпечення, особливо для таких категорій службовців, як мобільні співробітники, відряджені або надомні працівники. При цьому, флеш-накопичувачі з перебудовуваним ключем аутентифікатором мають ряд переваг при вирішенні задач комп'ютерної безпеки, зокрема комп'ютерної стеганографії.

Також, досить важливими є проведення порівняльної характеристики програмних засобів, що приховують інформацію (табл. 1.2).

JSTEG є однією з програм для вбудовування повідомлень в зображення формату JPEG. Алгоритм роботи її полягає у заміни найменш значущих біт

(метод LSB). Програма стійка до візуальних атакам і має високу пропускну здатність. Приховане повідомлення може займати до 12% всього зображення.

Таблиця 1.2 – Порівняльна характеристика програмних засобів, що приховують інформацію

Програмні засоби	Недоліки	Переваги
JSTEG	Не стійка до аналізу гістограм; не стійка до візуальних атакам; підтримує тільки формат JPEG; тільки один алгоритм шифрування; мало функціональних можливостей.	Повністю доступна програма для шифрування в JPEG; простота використання.
S-Tools	Підтримує мало форматів; мало функ- ціональних можливос- тей; спостеріється спотворення інформації що вбудовується.	Підтримка декількох алгоритмів шифрування; без пароля неможливо встановити факт роботи STools.
TrueCrypt	Виявлені серйозні уразливості і порушення, внаслідок чого проект закритий.	Створення віртуального зашифрованого диску; присутнє шифрування; використання хеш-функції при шифруванні; наявність двох рівнів захисту від виявлення; не вимагає встановлен- ня; є можливість резерв- ного копіювання даних.
CyberSafe	Є платною і вимагає спе- ціального встановлення.	Широкий спектр можливостей; необмежена довжина пароля для шифрування; використання алгоритмів шифрування; простота використання.
Folder Lock	Наявність тільки англійської мови; відсутня цифровий підпис; висока вартість.	Простий і зрозумілий інтерфейс; прозоре шифрування «на льоту».

S-Tools програма дозволяє приховувати будь-які файли в GIF, BMP і WAV. Здійснює регульований стиск (архівування) даних, крім того, виконує шифрування з використанням алгоритмів MCD, DES, потрійний-DES, IDEA. При шифруванні використовує пароль користувача.

TrueCrypt – це комп'ютерна програма шифрування і приховування файлів, для 32 і 64-розрядних операційних систем сімейств Microsoft Windows NT 5 і новіше (GUI-інтерфейс), Linux і Mac OS X. Дозволяє шифрувати логічні диски, розділи жорсткого диска або usb накопичувача.

Результат роботи програми представляється файлом, що містить всю зашифровану інформацію (каталоги, папки і т.д.).

CyberSafe надає широкий спектр послуг таких як: шифрування і приховування файлів, папок, каталогів і дисків, електронної пошти і т.д. Програма є платною і необхідно придбати ліцензію на право використання.

До основних можливостей Folder Lock відносяться: Aes-шифрування (довжина ключа 256 біт), приховування файлів і папок, шифрування файлів «на льоту», резервне копіювання-онлайн, створення захищених usb/cd/dvd-дисків, шифрування вкладень електронної пошти.

Таким чином, поряд з широко поширеними засобами захисту, що базуються на методах криптографії, відомі методи стеганографічної захисту інформації, основним завданням яких є приховування самого факту існування або передачі зашифрованих даних за допомогою носіїв, які не викликають підозр у зловмисника. Використання стеганографічних засобів на базі апаратно-програмних пристроїв персоналізації інформації, що дозволяють приховати факт роботи з такими програмами, забезпечує суттєве посилення криптографічного захисту даних як для корпоративних персональних комп'ютерів, так і для персональних комп'ютерів, що належать фізичним особам.

2.2 Методика і алгоритми приховування тексту в зображенні

Методика приховування інформації описує порядок приховування довільного тексту в кодуванні windows-1251 в 24-розрядний bitmap рисунка.

Відомо, що структура BMP-файлу складається з чотирьох частин:

- заголовок файлу;
- заголовок зображення (може бути відсутнім);
- палітра (може бути відсутнім);
- саме зображення.

Заголовок файлу містить службову інформацію, в тому числі розрядність рисунка. Для 24-розрядної рисунка палітра не використовується. Поскільки в завданні позначили, що необхідно працювати тільки з 24-

розрядними зображеннями, то можна перевіряти вхідне зображення на відповідність вимогам (рис. 2.1).

Перейдемо до самого зображення. Відомо, що формат BMP за замовчуванням не передбачає стиснення. Хоча на практиці існує підтримка стиснення по алгоритму RLE, але його не будемо використовувати в роботі.

Рисунок 2.1– Структура BMP-файлу

Таким чином, кожен піксель в нашому випадку кодується 24 бітами, по байту на кожну компоненту кольору. Отже, можна закодувати ні більше ні менше, а рівно 16777216 кольорів. Для наочності наведемо рисунок (рис. 3.2):

Заголовок файлу	Заголовок зображення	Палітра	Пікселі
14 байт	40 байт	Розмір залежить від кількості кольорів	Число байт визначається розмірами растра

Рисунок 2.2 – Формат BMP файлу за замовчуванням

Основна ідея методики приховування тексту полягає в тому, що око середньостатистичного якої людини розрізняє не багато кольорів, менш ніж 10 мільйонів. Звідси випливає те, що кілька молодших бітів з восьми, що відводяться на кожну компоненту кольору, можна запозичити для мети схову інформації.

Тому, під час роботи алгоритму зі схову інформації віднімається у RGB компонент по три молодших біта. Тобто з 24 біт палітри кольорів залишиться 18, якими можна закодувати рівно 262144 кольорів. Потім береться текст в кодуванні windows-1251, де кожен символ кодується 8 бітами. Так як 3 символа в форматі windows-1251 будуть складати 24 біта, то в 4 точках рисунка будемо економити 24 біта (так як 6x4=24). Тому, щоб сховати 3 символа в форматі windows-1251 буде достатньо 4 точок у форматі BMP. Ефективність такого приховування буде складати 25%.

Таким чином, якщо в 24-бітному зображенні формату BMP за стандартом 1024x768 знаходиться 786432 пікселів, то можна приховати текст у 196608 пікселях загальним розміром 147456 символів.

Алгоритм Jsteg – передбачає розмір прихованого повідомлення – 12,8% від об'єму контейнера, що є досить великим обсягом [11]. Він використовує заміну найменш значущого біта в зображеннях JPEG формату, з втратами при стисненні. Працює Jsteg наступним чином – замість молодшого біта вставляє біт повідомлення. І навіть при тому, що візуально різницю побачити майже не можливо, статистично виявити приховану інформацію доволі легко. Справа в тому, що Jsteg вносить певну бітову залежність в частоту появи найменш значущого біта. Тим самим впливає на пари частот рис 2.3 .

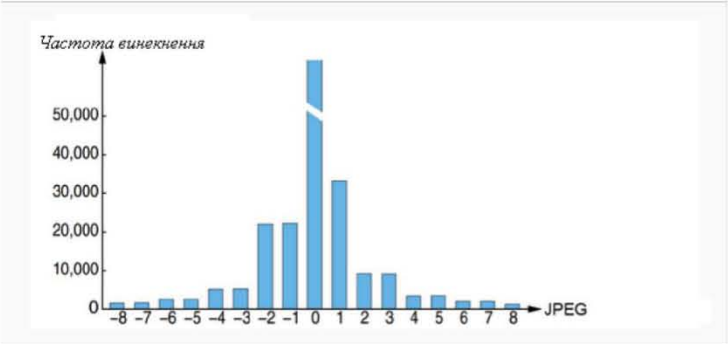

Рисунок 2.3 – Порівняння пар коефіцієнтів.

Припущення для зміненого зображення полягає в тому, що суміжні частоти однакові. Для визначення очікуваного розподілу обчислюється середнє арифметичне. Отримане значення порівнюється з емпіричним розподілом. Рисунок 2.2 ілюструє статистичну атаку на Jsteg стеганограму (з 50% заповненням контейнера, тобто 7680 байт). Діаграма являє ймовірність впровадження [12]:

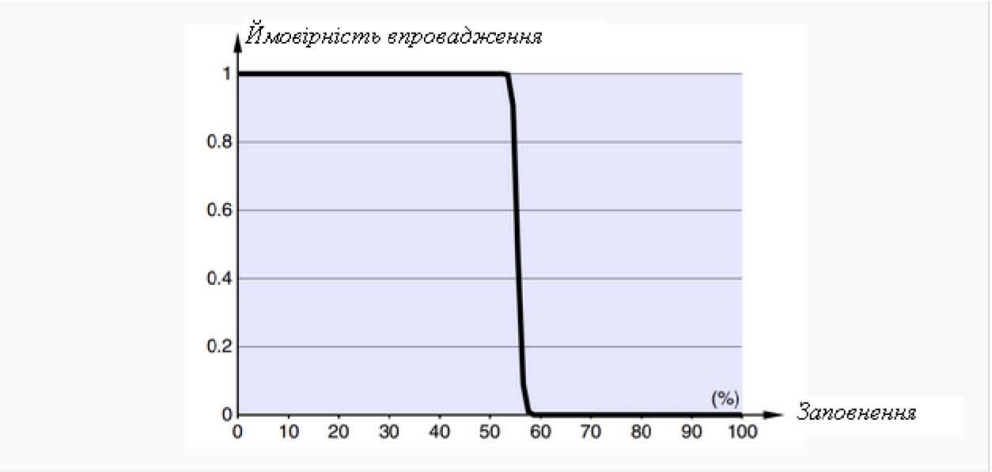

Рисунок 2.4 – Ймовірність впровадження в Jsteg стеганограмму при 50% заповнення.

Алгоритми стиснення зображень поділяють на стиснення без втрат – коли в процесі стиснення не порушується цілісність даних зображення, і стиснення з втратами – коли часткова втрата обумовлена методом який використовується. Відповідно, якщо файл призначений для довгострокового зберігання, то слід користуватись стисненням без втрат, так як це забезпечить повне відновлення вихідного зображення. Якщо ж зображення призначене суто для візуального сприйняття, то можливе використання стиснення з втратами, тому що вихідний сигнал і так має ймовірність вмісту шумів, але вони незначні і для спотворення зображення в цілому їх буде замало. Хоча загальний критерій оцінки якості зображення при втратах не висвітлений [13].

Алгоритм RLE – кодування з змінною довжиною рядка. Полягає в пошуку однакових пікселів в одному рядку, які заміняють парами. Призначений для роботи з діловою графікою, схемами, малюнками. Тобто де є області повторюваного кольору. Але якщо використовувати RLE в

кольорових зображеннях, це може призвести до збільшення розміру файлу. Продемонструвати роботу даного алгоритму можна наступним чином:

Вхідний потік даних:

17 8 54 0 0 0 97 5 16 0 45 23 0 0 0 0 0 3 67 0 0 8

Потік даних після кодування:

17 8 54 0 3 97 5 16 0 1 45 23 0 5 3 67 0 2 8

В основному для кодування застосовують схему під назвою PackBits, яка передбачає заміну кожних 7 бітів на 8, а наступний приймає значення прапора стиснення.

Вхідні дані:

1,2,3,4,2,2,2,2,4

Дані після кодування:

1,2,3,4,2, & 3,4

Формати в яких використовується RLE – BMP, TIFF, GIF.

Алгоритм Хаффмана зсилається на те, що окремі значення сигналу повторюються частіше і це можна використати для стиснення зображення [14]. А саме, користуватись меншим числом біт для збереження значень інтенсивності чим кількість самих значень. Питання лише в тому, що на різні значення виділяється своя кількість біт, тому їх потрібно якимось чином розділяти [15].

Потік даних після кодування:

(0010 0001 000011 1 0011 000010 01 0001 1)

Груповані блоки байтів:

(0010 0001) (000011 1 0) (011 00001) (0 01 00 1 0 1)

Метод стиснення Хаффмана можна проілюструвати так, як показано у таблиці 2.2.

Створені унікальні коди, які записуються в потік даних без зайвих маркерів. Програма відновлення визначає значення по кількості 0 до і після 1. Даний метод застосовують при кодуванні послідовності значень, що є складніша форма відносно кодування одного значення. Використовується алгоритм Хаффмана у форматах TIFF, GIF.

Таблиця 2.2 – ілюстрація методу стиснення Хаффмана

Значення	Частота згадки	Код Хафмана
A	.154	1
B	.110	01
C	.072	0010
D	.063	0011
E	.059	0001
F	.015	000010
G	.011	000011

Алгоритм LSB – полягає в вкраплені інформації в останні біти зображення, точніше їх заміни. Відбувається цей процес за допомогою кодування кольору на біти необхідної інформації.

Перетворення тексту в байтову послідовність відбувається наступним чином. Зображення зберігається у вигляді матриці, яка містить для кожної точки зображення своє значення кольору. Кожен компонент каналу кольору RGB зберігається в одному байті, завдяки чому може набувати значення від 0 до 255, що в свою чергу відповідає 24-х бітній глибині кольору. В людського ока не має особливості розрізняти невеликі коливання кольору. Тому при такій глибині кольору, заміна в кожному з каналів найменш значущого біта спотворення будуть менше 1% [16].

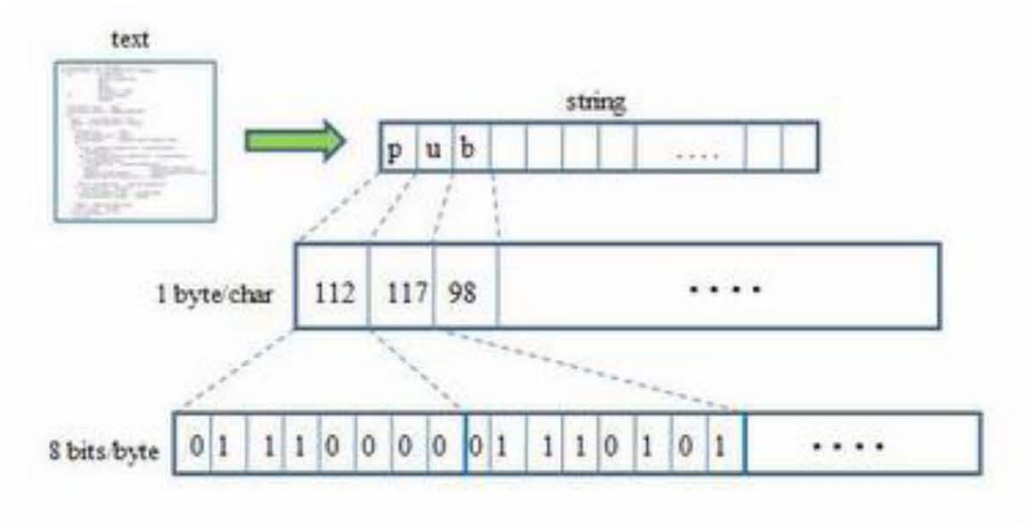

Рисунок 2.5 – Перетворення в байтову послідовність

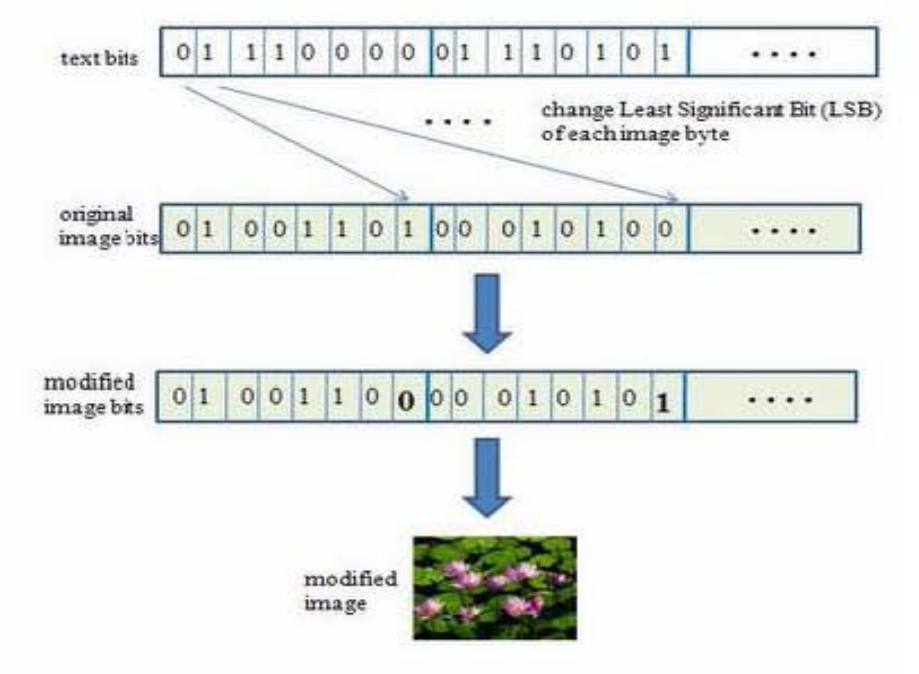

Рисунок 2.6 – Приховування інформації в зображенні.

Варто розуміти, якщо є методи приховування інформації, то є методи виявлення прихованої інформації. Одним з таких методів є RS-атаки. Метод ґрунтується на припущені, що для зображення яке є оригінальним, мається на увазі порожній контейнер. Розподіл буде таким самим як тоді коли значення пікселів будуть зсунуті на 1. Відбувається це так, зображення розбивається на групи по n пікселів, де n – парне число, припустим 2 пікселі. Вони горизонтально межують. Кожна група пікселів має функцію гладкості f(G), тобто суму перепадів значень суміжних пікселів [67-69]. Не заповнений контейнер буде мати співвідношення між групами з невеликими коливаннями. Якщо ж розбіжність між значеннями суттєва, то дане зображення було попередньо оброблене LSB-алгоритмом [70-72].

2.3 Розробка алгоритму приховування текстової інфомації

За основу даного алгоритму брався вже відомий LSB, але з певними модифікаціями. Для початку було запропоновано вбудовувати приховану інформацію не тільки в найменш значущі біти, а і в молодші. Це підвищило б обсяги приховуваного повідомлення. Але разом з цим виріс би відсоток спотворень і втрачалась суть використання стеганографічних методів, тому

що чим більше спотворень тим легше помітити факт передачі неозброєним оком.

Крім того є певні методи виявлення вкраплень інформації у мультимедійні файли. Наприклад, метод RS-атак або метод стегоаналізу Хі-квадрат.

Суть методу Хі-квадрата полягає в тому, щоб здійснювати атаку на один з каналів RGB, після чого привести обрахунки кожного з каналів до середнього арифметичного тим самим отримуючи результат вихідного зображення. Базується Хі квадрат на припущені того, що ймовірність появи сусідніх кольорів, які різняться між собою найменш значущим бітом, в порожньому стегоконтейнері – дуже мала. Тобто кількість пікселів двох сусідніх кольорів значно відрізняються для пустого контейнера і для виявлення необхідно всього лише порахувати кількість пікселів кожного кольору та застосувати кілька формул.

Нехай h – масив, на і-тому місці, який має кількість пікселів і-того кольору в досліджуваному зображенні.

Тоді: виміряна частота появи кольору i=2k:

$$n_k = h[2k],\ k \in [0, 127];$$

Теоретично очікувана частота появи кольору i=2k:

$$n_k^* = \frac{h[2k] + h[2k+1]}{2},\ k \in [0, 127];$$

Але слід пам'ятати, що застосування Хі квадрата буде більш ефективним, якщо це робити не до всього зображення, а до окремих його частин. Наприклад до стрічок. Таким чином, якщо обрахована ймовірність для стрічки буде більше 0.5, то в даній стрічці є вкраплена інформація.

Тому альтернативним варіантом залишився той самий LSB, але з зміненою послідовністю заміни найменш значущих бітів. За допомогою генератора псевдовипадкових чисел послідовність з лінійної стає довільною, що зводить можливість виявлення за допомогою методу Хі квадрата до мінімуму. Також покращена версія алгоритму приховування передбачає

додатковий захист у вигляді криптографічного алгоритму, який додає до приховування факту передачі повідомлення, його шифрування. І навіть в разі виявлення вкрапленого повідомлення потрібно буде розшифрувати його вміст.

Обраним алгоритмом став RSA (Rivest-Shamir-Adleman), фундаментом якого є властивості простих але дуже великих чисел. Простими числами називають ті в яких дільник або одиниця, або саме число. А взаємно простими числами є тільки ті в яких дільник лише одиниця.

Для створення ключа обирається два великих простих числа. Чим більші числа використовуються, тим більш крипто-стійкими вони будуть вважатись. Наприклад, UNIX-програма ssh-keygen генерує ключі довжиною 1024 біта за замовчуванням. Така довжина ключа є мінімальною для алгоритму RSA в силу його особливостей. А для асиметричних алгоритмів заснованих на теоріях при використанні еліптичних кривих, мінімально надійною довжиною ключа є 163 біта, хоча рекомендованою є довжина від 191 біта [17].

RSA алгоритм дозволяє шифрувати інформацію в кількох режимах:

- таємний ключ відправника, у такому разі повідомлення може розшифрувати будь-яка людина, яка має в наявності відкритий ключ.
- відкритий ключ отримувача, дає змогу дешифрувати повідомлення власнику таємного ключа, але процес дешифровки буде успішним при наявності відкритого ключа, тому що вони є парними.
- таємний ключ відправника та відкритий ключ отримувача повідомлення, тільки тоді повідомлення може бути розшифрованим на стороні отримувача.

В цьому варіанті модифікації передбачається використання саме третього варіанту шифрування.

Вивести закритий ключ з відкритого вкрай важко, якщо взагалі можливо. Для цього необхідно вирішити задачу розкладу дільників величезного цілого числа (необхідно розкласти на співмножники 129-значне число). До цих пір аналітичними методами вона не була вирішеною, тому вважається, що RSA можливо зламати лише шляхом грубого перебору. А до

того часу поки Bruteforce attack буде реалізовано, актуальність переданої інформації скоріше за все буде втрачено.

2.4 Висновок до розділу

В другому розділі проведено аналіз та порівняння програмних засобів, розглянуто методики та алгоритми приховування інформації, запропоновано покращений алгоритм з додатковим захистом. Також обґрунтовано вибір стеганографічного алгоритму над яким проведено покращення та описаний алгоритм шифрування заповненого контейнеру.

За допомогою даної комбінації алгоритмів відбувається уникнення вияву прихованого повідомлення методом Хі-квадрат і RS-атак та надається додатковий захист на випадок, якщо зловмиснику все ж вдалось виявити факт передачі.

3 РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ПРИХОВУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ

3.1 Розробка архітектури програмного продукту

При розробці програмного продукту використовувались наступні засоби:

- середовище програмування VS Code.
- HTML для розбиття веб-сторінки на логічні і зрозумілі браузеру блоки.
- SCSS для стилізації інтерфейсу.
- JS для додавання динамічності веб-сторінки.
- PHP функції, які призначені для взаємодії з віддаленим сервером.

Робота програми складається з таких кроків:

- запуск веб-додатку.
- додати файл-контейнер для вкраплення повідомлення;
- додати повідомлення, яке безпосередньо буде приховуватись в контейнер зображення.
- вписати таємний ключ відправника.
- відправка заповнених користувачем полів на сервер де буде здійснена обробка іншою програмою, робота якої базується на комбінації покращеного стеганографічного алгоритму LSB та алгоритму шифрування RSA.
- оброблені дані повертаються відправнику і автоматично починають завантажуватись на комп'ютер в новий файл який вже є заповненим контейнером.
- отриманий файл передати по раніше встановленому каналу зв'язку між відправником та отримувачем.

Інструкція користувачу.

- веб-додаток призначений для приховування текстових повідомлень у зображеннях;
- користувач взаємодіє з інтерфейсом через браузер, який в свою чергу використовує віддалений сервер для обробки даних, які надає користувач. Для роботи веб-додатку користувачеві не потрібно

виконувати ніяких системних налаштувань. Інтерфейс створений максимально простим та інтуїтивно зрозумілим;

- після того як користувач заповнить необхідні поля та відправить дані на сервер, почнеться обробка за вдосконаленим алгоритмом приховування. Після чого заповнений контейнер буде автоматично скачано на комп'ютер користувача;
- отриманий файл передати адресату по загально доступному або раніше встановленому каналу зв'язку;
- отримувачеві для дешифровки необхідно запустити веб-додаток і помістити у відповідне поле заповнений файл-контейнер;
- після чого ввести відкритий ключ та підтвердити дешифровку зображення;
- оброблені дані повернуться до отримувача та будуть розділені на дві області, в одній знаходиться не заповнений контейнер зображення, в іншій повідомлення яке було приховане.

3.2 Розробка програми приховування інформації

Як було зазначено вище, основним принципом стеганографії є передача секретної інформації в середині відкритої, яка не викликає підозри. Тобто приховання самого факту передачі. Наприклад один з типів інформації, текст, аудіо чи зображення вбудовується в середину іншого типу інформації. Таким чином контейнер (відкрита інформація) має звичайний вигляд. Одним з найбільш поширених методів який виконує приховування тексту в зображенні, реалізація у вигляді веб-додатків чи окремих програмних засобів.

Для розробки серверної частини веб-додатків, інтенсивно використовується мова програмування PHP. Структура файлу створюється за допомогою мови гіпертекстової розмітки. HTML залежно від потреби або дій користувача викликається скрипт чи стороння програма, для обробки даних.

Програмні рядки такого файлу, без дизайну мають вигляд:

```
Контейнер:
```

```
    <input name="!{ class }-image" type="file"
id="select-image__file" accept="image/x-png, image/gif,
image/jpeg">

    Повідомлення:
    <div id="encod-text">
            <textarea name="!{ class }-text" id="encod-
text__textarea" cols="30" rows="5" maxlength="64"
placeholder="!{ textarea_placeholder }"></textarea>
    </div>

    <button type="button" class="submit" id="btn-
submit">!{ button_text }</button>
```

В блок з назвою “Контейнер” завантажується зображення, яке буде використовуватись в ролі контейнера, а в полі з ідентифікатором “encod-text” записується повідомлення яке буде приховуватись в контейнер. Кнопка “btn-submit” відправляє дані з заповнених полів для подальшої обробки. А саме вбудовування текстової інформації в зображення та шифрування.

Також, часто під час розробки веб-додатків користуються фреймворком Ajax, точніше його функцією виклику. Тому до файлу index.php підключається фреймворк Ajax та виконується виклик скрипта або файлу js, наприклад stego.js:

```
<script src = "ajax.js"> </script>
<script src = "stego.js"> </script>
```

Якщо використання прямих та обернених алгоритмів приховування інформації виконуються у файлах stego.php і destego.php відповідно, то програмні рядки головного файлу index.php для виклику необхідних скриптів будуть виглядати наступним чином:

```
function stego (url, code) {
    $("#img_new").html("");
    $.ajax({
        type: "POST",
        url: "stego.php",
        data:
    'url_img='+url+'&stego_code='+code+'',
        cache: false,
    success: function(html){
    $("#img_new").html(html);
```

```
        }
        });
    }
```

Для реалізації оберненого алгоритму приховування інформації виконуємо у файлі destego.php наступним чином:

```
function de_stego (url) {
    $("#img_new").html("");
    $.ajax({
        type: "POST",
        url: "destego.php",
        data:
'url_img='+url+'&stego_code='+code+'',
        cache: false,
        success: function(html){
$("#img_new").html(html);
        }
    });
}
```

Обидві функції ініціалізують блоки, які відправлють відповідні дані з полів, введених користувачем до програми на сервері, де і відбувається обробка. Після всіх процесів заповнений контейнер завантажується на комп'ютер.

На початку роботи є кнопки, при натисканні на які обирається потрібний сценарій. Їх всього два, – приховати текст "set text", вилучити текст "get text".

```
    <section class="buttons">
          <button type="button" class="btn-action"
id="get-btn" data-action="get">Get text</button>
          <button type="button" class="btn-action"
id="set-btn" data-action="set">Set text</button>
        </section>
```

Для того, щоб вбудувати повідомлення до зображення необхідно виконати перевірку наявності зображення за адресою та відповідність формату графічного файлу (jpg, gif або png) за допомогою коду:

```
    if (! $ _ POST [url_img]) {echo "Введіть всі дані";
exit (); }
```

Якщо ж файл не може бути оброблений по причині хибного формату зображення, браузер виведе інформаційне повідомлення з помилкою. Якщо ж графічний файл коректний, він буде завантажений для вбудовування, але відсоток схову залежний від розширення зображення. Попередня перевірка розширення файла-контейнера може відбуватись наступним чином:

```
    $type_img = explode (".", $_POST[url_img]);
    switch(strtolower($type_img[count($type_img)-1])) {
    case "png": $img1 =
imagecreatefrompng($_POST[url_img]); break;
    case "jpg": $img1 =
imagecreatefromjpeg($_POST[url_img]); break;
    case "jpeg": $img1 =
imagecreatefromjpeg($_POST[url_img]); break;
    case "gif": $img1 =
imagecreatefromgif($_POST[url_img]); break;
    default: $img1 = imagecreatefromgd
($_POST[url_img]); break;
    }
```

Необхідно звернути увагу на те, що при підборі хостингу, є ймовірність того, що деякі сервіси можуть не включати в пакеті послуг наявність графічної бібліотеки GD у PHP. Це може призвести до збоїв або припинення роботи веб-додатку.

Зображення користувача можна вивести на екран у форматі png. Вибір саме цього формату полягає у тому, що з цим форматом краще порівнюються зображення між собою:

```
    imagepng ($ img1, "img / img_code1" .date (si).
".png");
    echo "Вхідний зображення: <img src =
'img/img_code1" .date (si).".png' width = '400px'>";
```

Визначити розмір зображення та задати початкові координати *x*, *y*, треба для того, щоб дізнатись довжину повідомлення. Далі додаєм повідомлення в робочу змінну *$code* наступним чином:

```
$Size = getimagesize ($ _ POST [url_img]);
$W = $ size [0];
$H = $ size [1];
$X = $ y = 0;
$Length = strlen($_POST[stego_code]);
```

```
$Code = $_POST[stego_code];
```

Цикл в ході якого текст потрапить до файла-контейнера програмно може мати вигляд:

```
while ($length--) {
$Color_pixel = imagecolorat($img1, $x, $y);
$Color_pixel_RGB = imagecolorsforindex($img1,
$color_pixel);
$Color_pixel_RGB[blue] = ord($code[$length]);
$Color_new_pixel = imagecolorclosest($img1,
$color_pixel_RGB[red], $color_pixel_RGB[green],
$color_pixel_RGB[blue]);
imagesetpixel ($img1, $x, $y, $color_new_pixel);
$X + = 50;
if ($x > $w) {$x = 0; $Y++;}
}
```

Даний цикл повторюватиметься до тих пір, поки не закінчеться повідомлення. Крім того, початок його буде з останнього символу. В процесі обробки дізнаємося колір пікселя за координатами x та y функції:

```
$color_pixel = imagecolorat($img1, $x, $y);
```

Значення співвідношення каналів кольорів в пікселях, заносим в масив *$color_pixel_RGB*. Програмний рядок:

```
$color_pixel_RGB[blue] = ord($code[$length])
```

таким чином визначається рівень синього кольору який відносно позиції в ASCII таблиці замінюється на номер символу повідомлення.

Зручно це тому, що вони повинні бути в діапазоні від 0 до 255 включно як і значення відповідно каналів кольору. Обирається синій колір, хоча можна спробувати з зеленим або червоним. Програма не використовує зелений чи червоний колір, але є виняток який описаний нижче.

У циклі наноситься піксель на зображення та збільшення координат. Якщо вони перевищують ширину зображення, то відбувається збільшення значення координати по висоті (y), а x прирівнюється до нуля.

Потім, виводимо нове зображення на екран браузера:

```
    imagepng($img1, "img/img_code2" .date (si).
".png");
    echo "Початкове зображення: <img src =
'img/img_code2" .date (si). ".png' width = '400px'>";
```

Таким чином виконується приховування інформації в зображенні.

Загальний вигляд файлу stego.php розміщений у додатку А.

Коли останій піксель тексту змінює червоний колір на 1, тоді скрипт розуміє, що це кінець повідомлення в процесі вилучення інформації. Коди виглядає наступним чином:

```
    $color_new_pixel = imagecolorclosest($img1, 1,
$color_pixel_RGB[green], $color_pixel_RGB[blue]);
    imagesetpixel($img1, $x, $y, $color_new_pixel);
```

Невід'ємною частиною процесу приховування – є відновлення інформації.

Початок коду для відновлення повідомлення може мати вигляд:

```
    if (!$_POST[url_img]) {echo "Введіть всі дані";
exit (); }
    $Type_img = explode (".",$_POST[url_img]);
    switch(strtolower($type_img[count($ type_img)-1]))
{
    case "png": $img1 = imagecreatefrompng($
_POST[url_img]); break;
    case "jpg": $img1 =
imagecreatefromjpeg($_POST[url_img]); break;
    case "jpeg": $img1 =
imagecreatefromjpeg($_POST[url_img]); break;
    case "gif": $img1 =
imagecreatefromgif($_POST[url_img]); break;
    default: $img1 =
imagecreatefromgd($_POST[url_img]); break;
    }
    echo "Вхідне зображення: <img src = '".
$_POST[url_img]."' width = '400px'>";
    echo "Код:";
    $Size = getimagesize ($ _ POST [url_img]);
    $W = $size[0];
    $H = $size[1];
    $X = $y = 0;
```

В наведеному прикладі коду програми здійснюється перевірка введеної інформації, вивантаження цифрового контейнера і виведення його у вікні браузера. Після чого йде цикл дешифрування інформації:

```
    while ($color_pixel_RGB[red]!=1)  {
    $color_pixel = imagecolorat ($img1, $x, $y);
    $color_pixel_RGB   =   imagecolorsforindex   ($img1,
$color_pixel);
    $text = chr($color_pixel_RGB[blue]).$text;
    $x+=50;  if ($x>$w) {$x=0; $y++;}
    } ;
```

Цикл буде працювати поки не дістанеться до позначеного місця в повідомленні, яке є його кінцем. Таким чином пікселі будуть перевірятись і отримувати символи в тому ж порядку в якому вносилось повідомлення. Вивід повідомлення:

```
$text[0] = "";
echo $text;
```

Програмний рядок *$text[0] = ""*– ініціалізує строкову змінну.

3.3 Перевірка працездатності програми

Інтерфейс веб-додатку виглядає наступним чином:

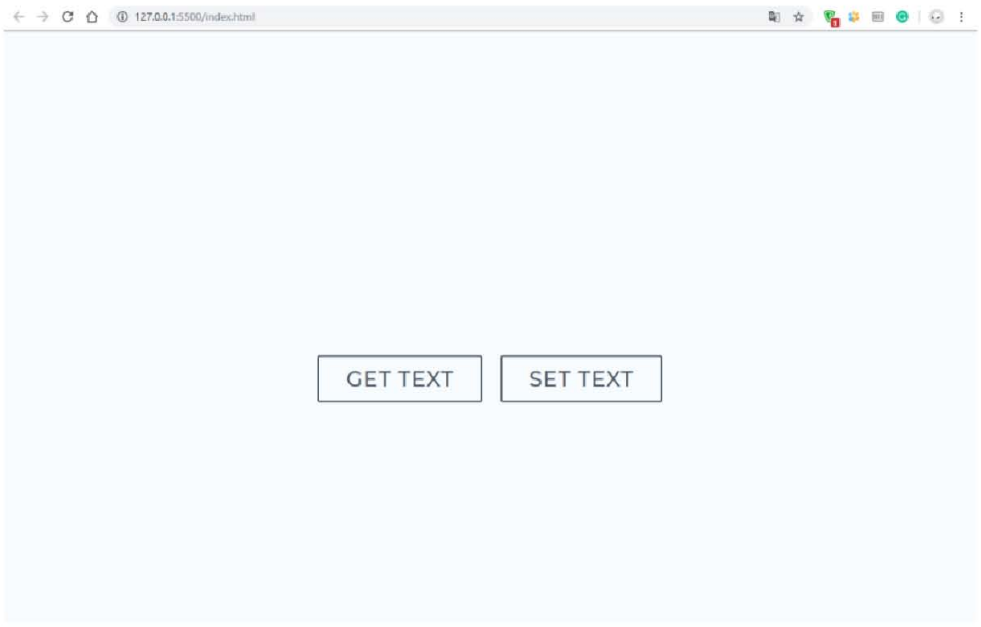

Рисунок 3.1 – Початок роботи з веб-додатком.

В наявності дві кнопки ‘Get text’ – вилучити текст та ‘Set text’ – додати текст. Для вкраплення повідомлення в зображення обираєм ‘Set text’.

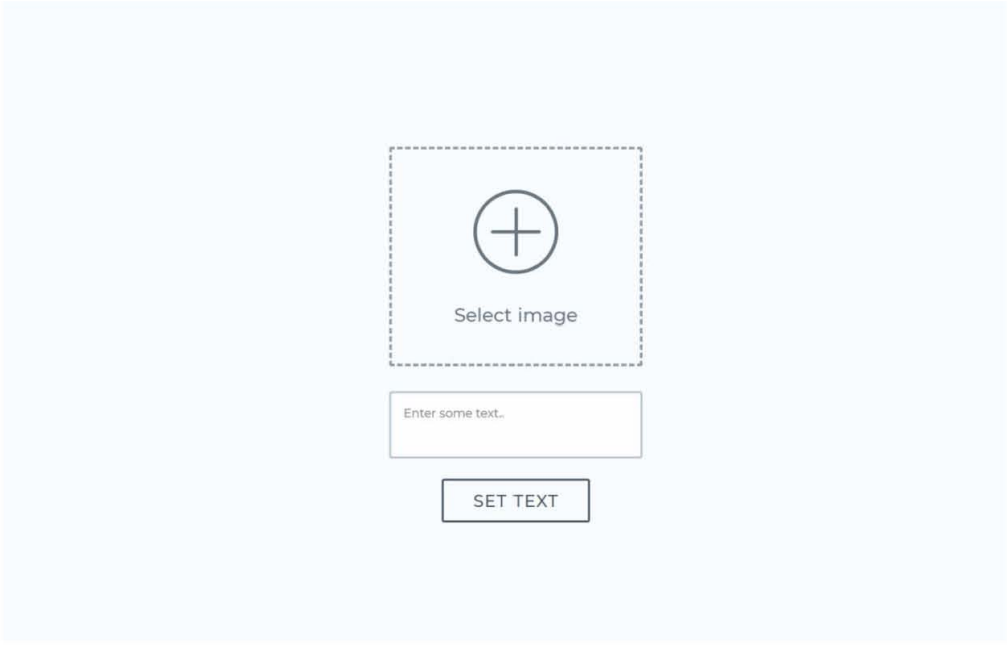

Рисунок 3.2 – Завантаження контейнера.

Після чого бачимо область в яку необхідно завантажити зображення, поле для повідомлення яке потрібно приховати та кнопку для відправки даних з полів на сервер для обробки.

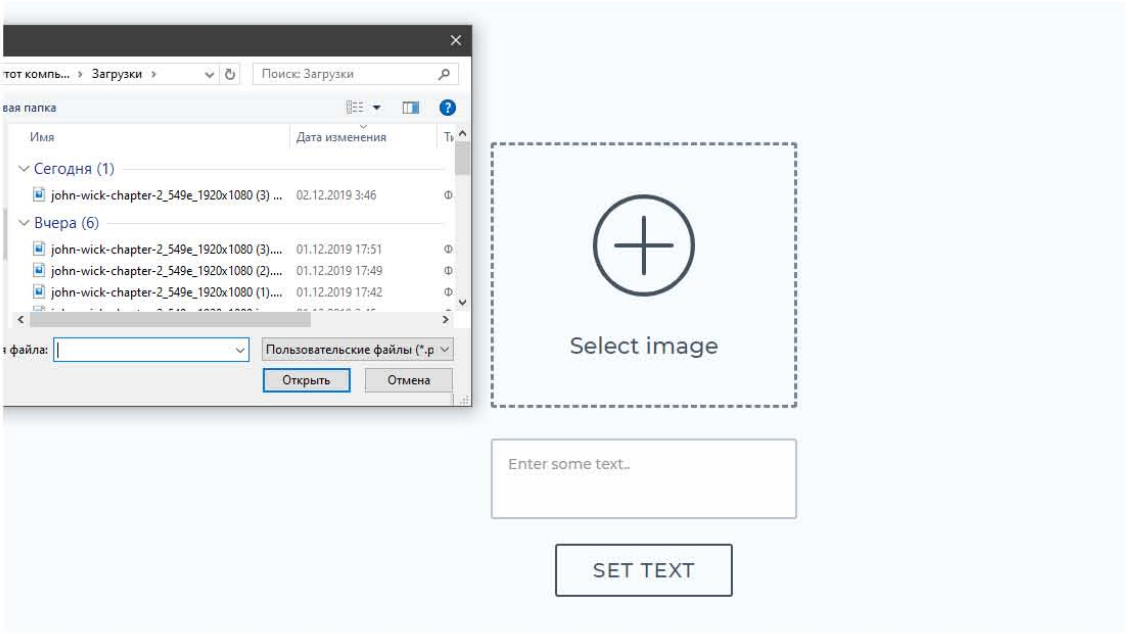

Рисунок 3.3 – Вибір контейнера для приховування.

Наступним кроком є саме вивантаження незаповненого файлу-контейнера.

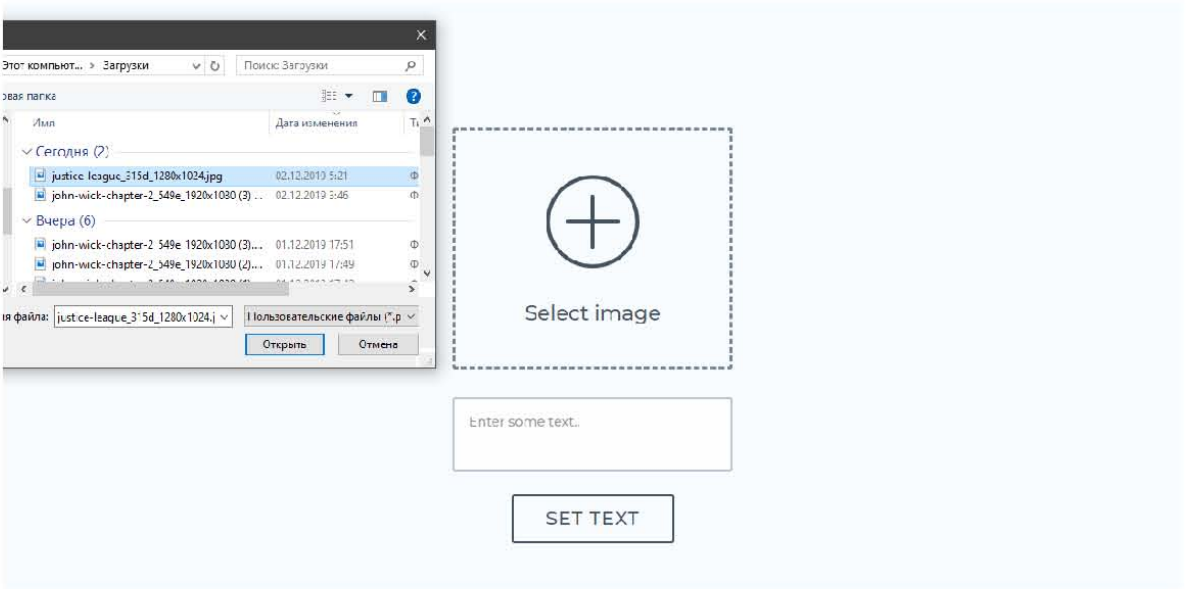

Рисунок 3.4 – Завантаження контейнера.

Обране зображення буде показано у відповідній області. У текстовому полі, додаєм повідомлення для вкраплення. І тиснем кнопку “set text” – вбудувати текст.

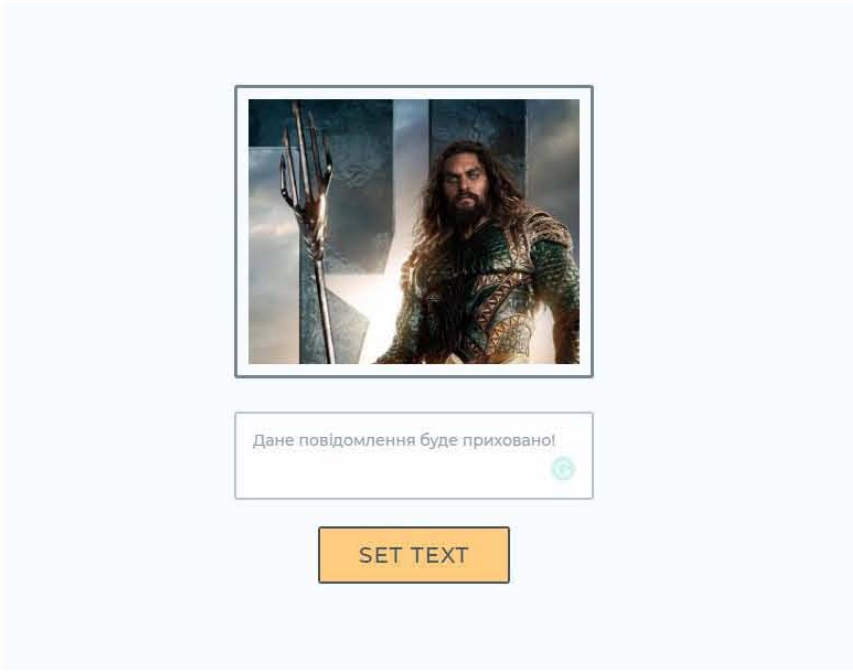

Рисунок 3.5 – Відправка даних для обробки.

Фон кнопки отримує статус “success”, який означає що вбудовування пройшло успішно.

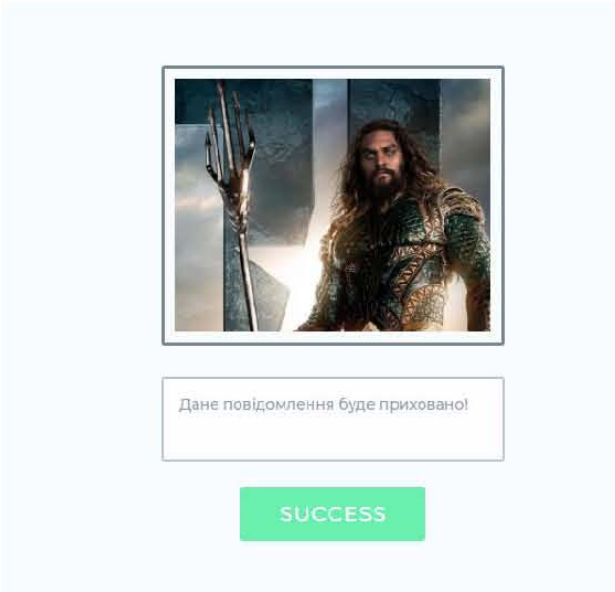

Рисунок 3.6 – Коректна відправка даних.

Одразу після цього почнеться автоматичне завантаження заповненого контейнеру на комп’ютер користувача.

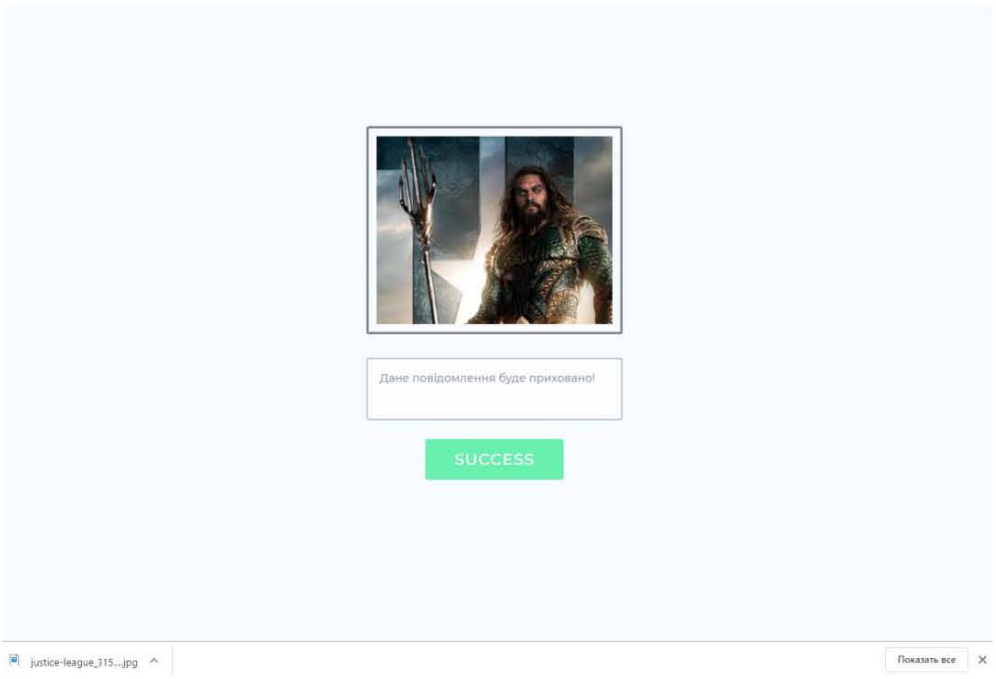

Рисунок 3.7 – Завантаження заповненого контейнера.

Для зручності користування, щоб не виникало необхідності перезавантажувати сторінку, до інтерфейсу веб-додатку було створено кнопку “refresh” – повернутись на початок роботи.

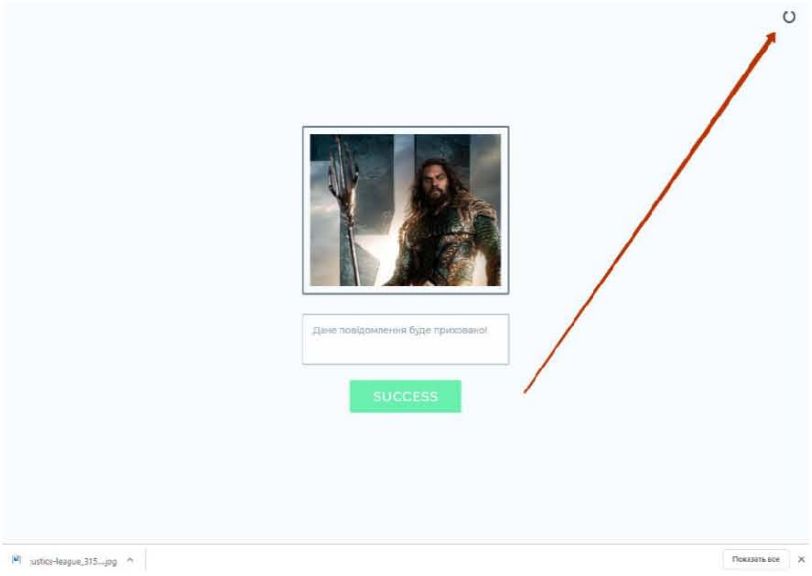

Рисунок 3.8 – Кнопка повернення до початку роботи.

Для вилучення даних з зображення спочатку потрібно перейти з початкового стану веб-додатку до вкладки “get text” – вилучити текст.

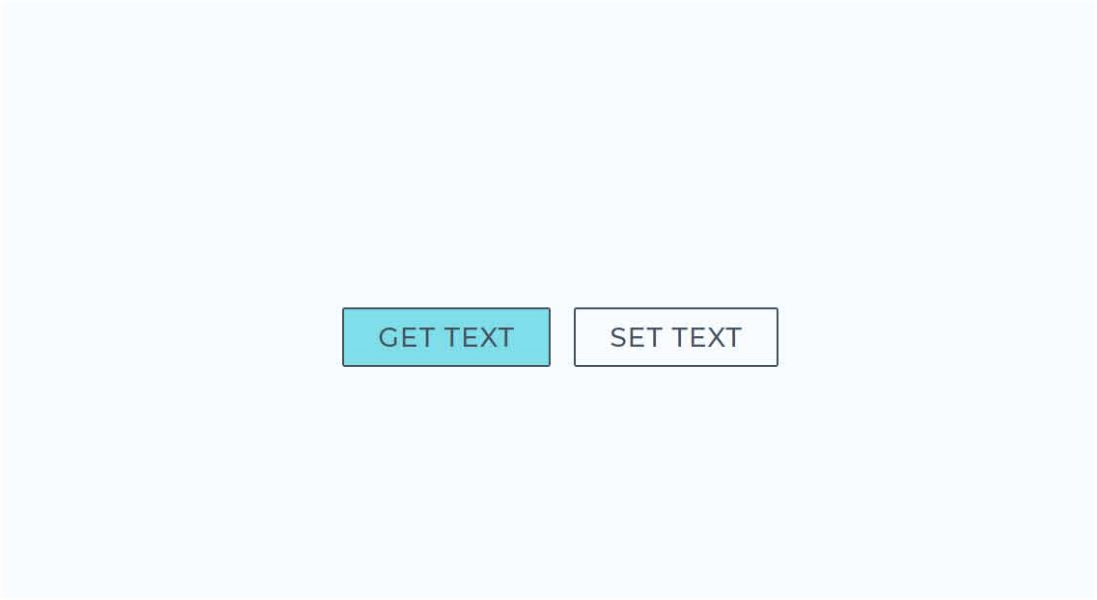

Рисунок 3.9 – Вилучення інформації.

В область для зображення необхідно завантажити заповнений контейнер. І натиснути кнопку “get text”.

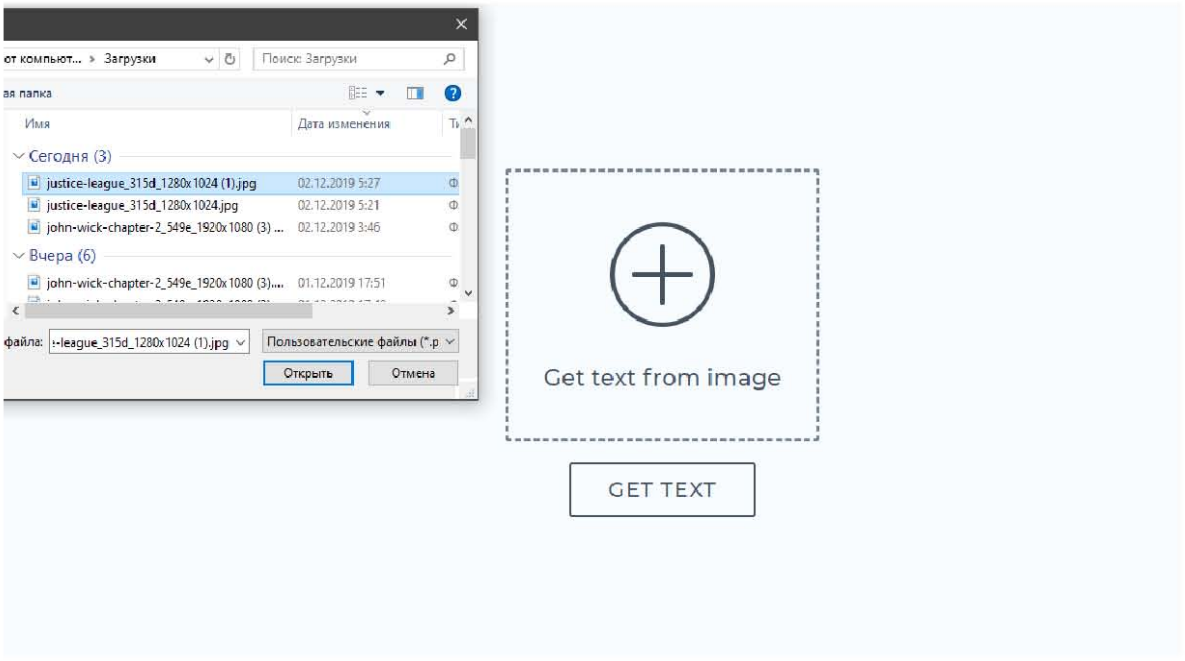

Рисунок 3.10 – Вибір заповненого контейнера.

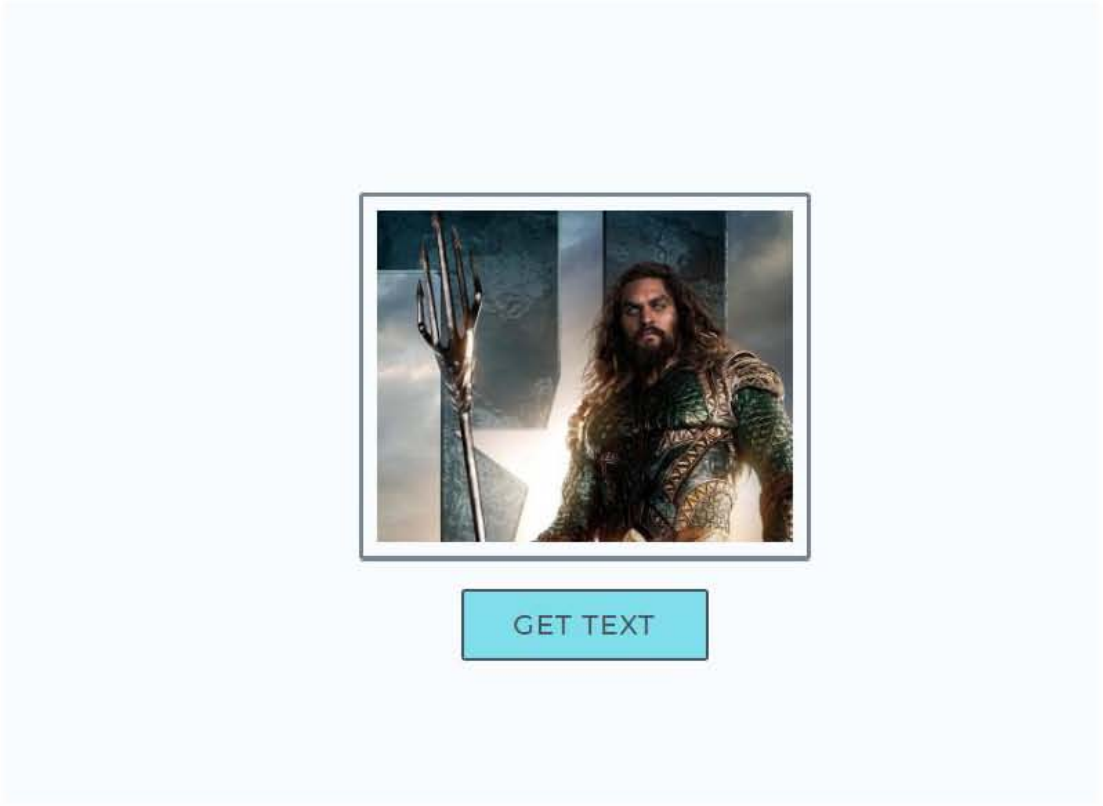

Рисунок 3.11 – Контейнер з вкрапленою інформацією.

Після чого з’явиться поле з раніше вбудованою інформацією.

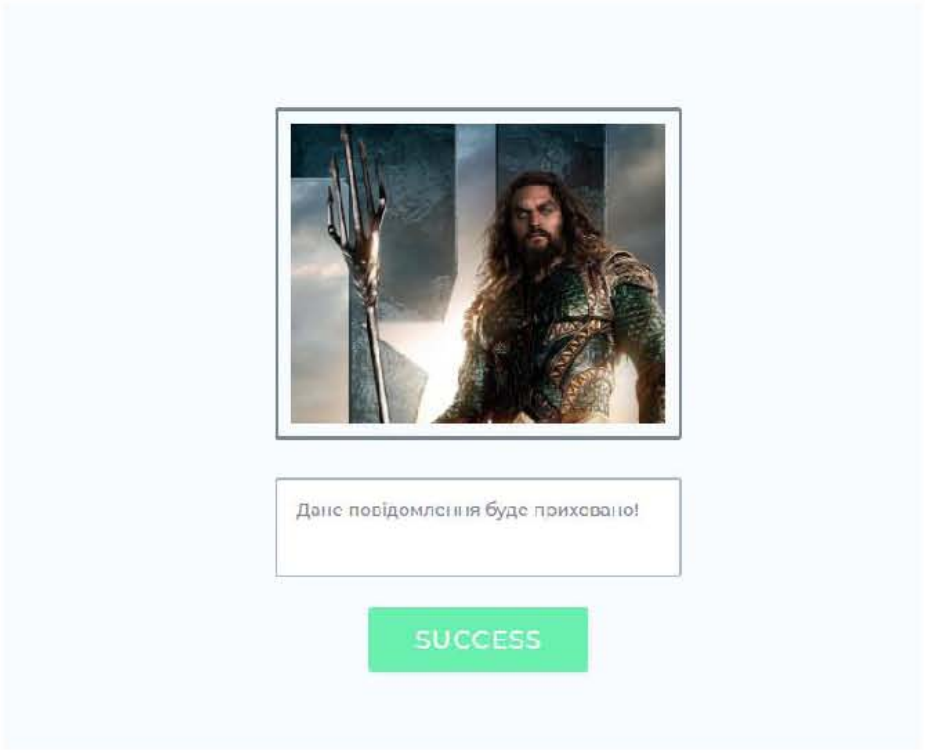

Рисунок 3.12 – Виявлення прихованого повідомлення.

При спробі завантажити не заповнений контейнер, навколо зображення з’явиться червона рамка, яка буде означати те, що зображення не містить ніяких вбудованих даних.

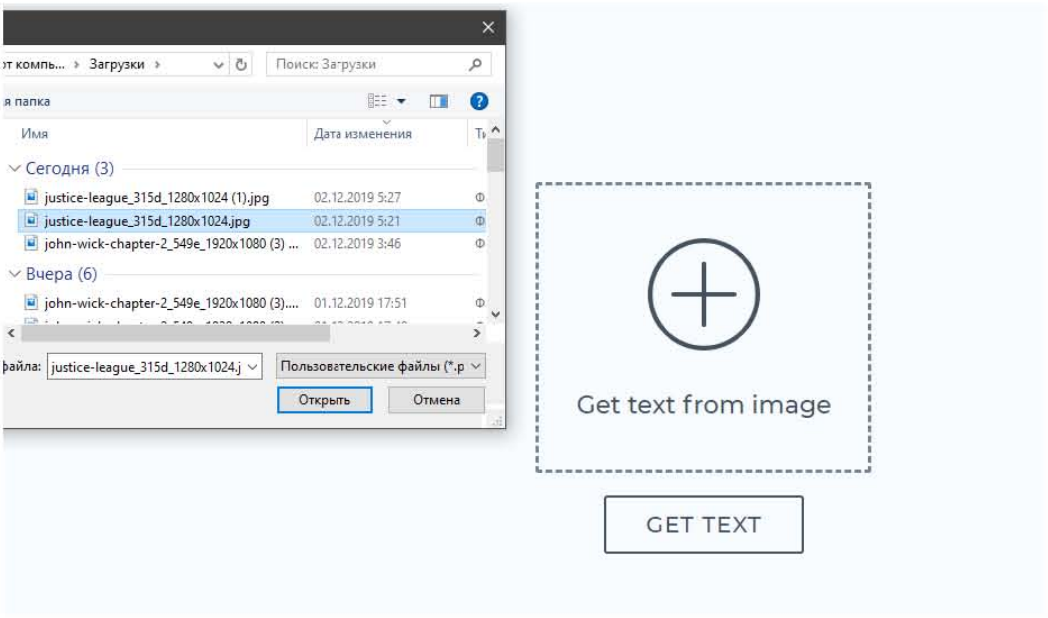

Рисунок 3.13 – Вибір порожнього контейнера дял вилучення інформації.

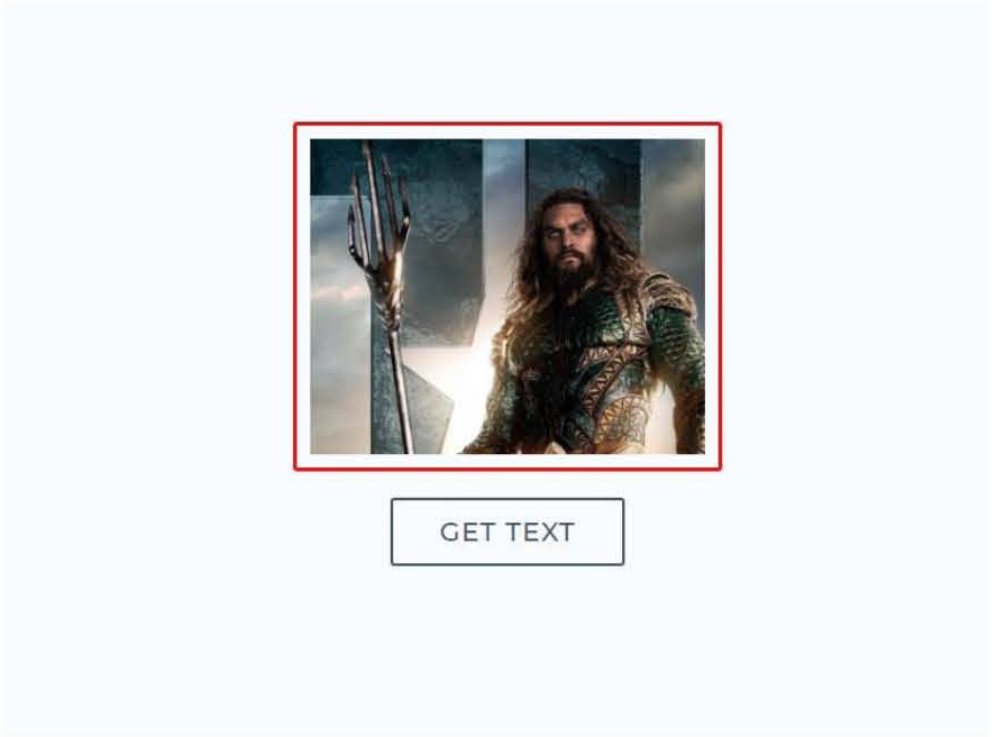

Рисунок 3.14 – Не коректний контейнер.

3.4 Висновок до розділу

У межах розділу описано архітектуру програмного продукту, які засоби були використані для створення веб-додатку, з яких кроків складається робота програми. Створена інструкція користувачу, описані частини коду для роботи програми, приведена робота веб-додатку у графічному вигляді. Під час тестування збоїв не було. Обрана комбінація методів передбачає безпечніше приховування тексту, стійкість до вияву факту вбудовування, додатковий захист у вигляді шифрування. Інтерфейс є максимально простим та інтуїтивно зрозумілим.

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ СТВОРЕННЯ ПРОГРАМИ ПРИХОВУВАННЯ ТЕКСТОВОЇ ІНФОРМАЦІЇ У ЦИФРОВОМУЗОБРАЖЕННІ

4.1 Загальні положення

У сучасних умовах відсутність впровадження інновацій в промисловості відчувається особливо гостро. Підприємства потребують швидкого впровадження досягнень науково-технічного прогресу галузі, а це звісно вимагає зменшення часу на проведення науково-дослідних робіт і скорочення строку окупності витрат. Виходячи з цього, доцільно орієнтуватися на час проведення науково-дослідних робіт та розробку експериментального зразка продукту не більше 1 року, при чому технічні показники результатів плануються на рівні кращих світових зразків; термін окупності витрат у межах 1-2 років і менше. Впродовж подальшого розвитку роботи витрати повинні поступово зменшуватися, це свідчитиме про успішну реалізацію продукту і його своєчасне удосконалення.

На основі економічних розрахунків можна довести економічну доцільність та ефективність впровадження отриманих результатів виконаних науково-дослідних робіт у виробництво, тобто здійснити так звану комерціалізацію наукових розробок [18].

Саме цим завданням присвячено даний розділ магістерської кваліфікаційної роботи і передбачає він виконання таких етапів робіт (рис. 4.1):

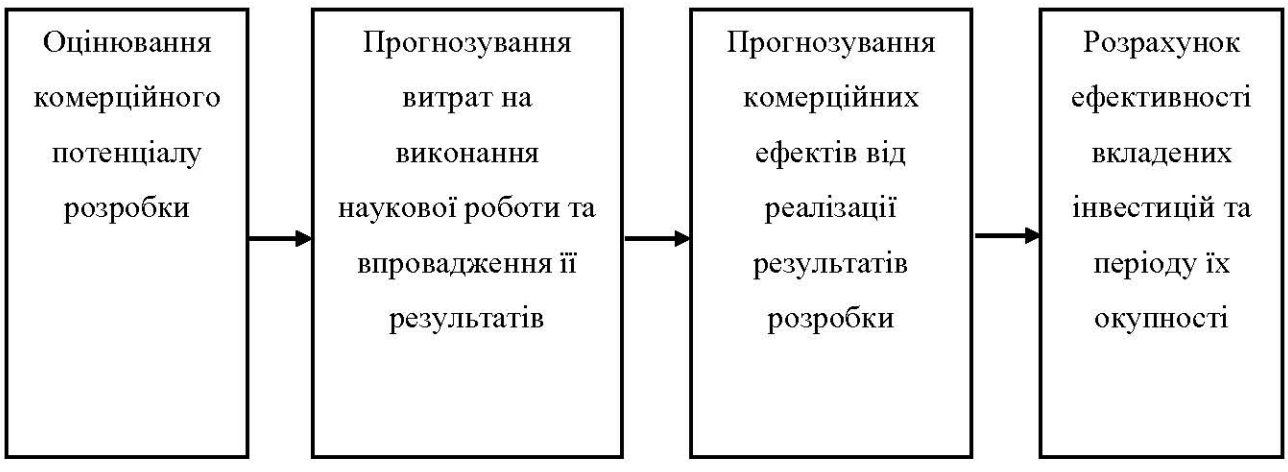

Рисунок 4.1 - Складові економічної частини магістерської кваліфікаційної роботи

Саме на такі складові буде поділено економічну частину даної магістерської роботи. Усі подальші економічні розрахунки, будуть висвітлені у згаданих підрозділах економічної частини. У комплексі ці етапи дозволять побачити цілісну картину доцільності розробки та впровадження запропонованого рішення.

4.2 Оцінювання комерційного потенціалу розробки

Метою проведення оцінювання комерційного потенціалу розробки є оцінювання комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності. В результаті оцінювання робиться висновок щодо напрямів (особливостей) організації подальшого її впровадження з врахуванням встановленого рейтингу.

Оцінювання комерційного потенціалу розробки будемо здійснювати за 12-ма критеріями, наведеними в таблиці 4.1.

Таблиця 4.1 - Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено працездатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Багато аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів

Продовження таблиці 4.1

5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років.	Термін реалізації ідеї менше 3-х років.	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідно регламентні документи документів на виробництво та реалізацію продукту	Необхідно отримання дозвільних документів на виробництво та реалізацію продукту	Отримання дозвільних документів для реалізації вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

На основі складеної таблиці ряд незалежних експертів, у нашому випадку керівник магістерської роботи та викладачі випускової кафедри виставили різні бали. Результати цього оцінювання комерційного потенціалу занесено до таблиці 4.2.

Таблиця 4.2 - Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	1СавицькаЛ.А., к.т.н., доц. кафедри ОТ	2 Семеренко В.П., к.т.н., доц. кафедри ОТ	3 Богомолов С.В., к.т.н., доц. кафедри ОТ
	Бали, виставлені експертами:		
1	3	3	3
2	2	2	2
3	2	3	2
4	3	2	2
5	3	2	2
6	3	4	3
7	0	1	0
8	3	3	3
9	1	1	1
10	4	4	4
11	3	4	3
12	3	3	3
Сума балів	$СБ_1 = 30$	$СБ_1 = 32$	$СБ_1 = 28$
Середньо-арифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{30+32+28}{3} = \frac{90}{3} = 30$		

За даними таблиці 4.2, а також згідно рекомендацій, що наведені в таблиці 4.3, можна зробити висновок, щодо рівня комерційного потенціалу розробки.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 - 10	Низький
11 - 20	Нижче середнього
21 - 30	Середній
31 - 40	Вище середнього
41 - 48	Високий

Взявши до уваги, середньоарифметичну суму балів, $\overline{СБ} = 30$, що були виставлені експертами, можна стверджувати, що рівень комерційного потенціалу даної розробки є середнім. Крім того слід зазначити:

- передбачається, що програмний продукт Double Protection буде реалізовуватись через спеціально розроблений сайт;
- новизною в даному випадку є покращена робота алгоритму, та доповнення його методом шифрування, комбінація алгоритмів та зміна роботи одного з них передбачає підвищення захисту, тому що це дозволить уникати кількох методів виявлення роботи алгоритму;
- розробка корисна тим, що дозволить захистити приватну інформацію від крадіжок;
- в даний час розробка знаходиться на стадії розробки промислового зразка;
- зацікавленою особою є будь-яка людина, яка має бажання захистити приватну інформацію.

Таблиця 4.4 – Порівняльна характеристика програмних засобів, що приховують інформацію

Програмні засоби	Недоліки	Переваги
JSTEG	Не стійка до аналізу гістограм; не стійка до візуальних атакам; підтримує тільки формат JPEG; тільки один алгоритм шифрування; мало функціональних можливостей.	Повністю доступна програма для шифрування в JPEG; простота використання.
S-Tools	Підтримує мало форматів; мало функ-ціональних можливос- тей; спостеріється спотворення інформації що вбудовується.	Підтримує кілька алгоритмів шифрування без пароля неможливо встановити факт STools.
TrueCrypt	Виявлені серйозні уразливості і порушення, внаслідок чого проект закритий.	Створення віртуального зашифрованого диску; присутнє шифрування; використання хеш-функції при шифруванні; наявність двох рівнів захисту від виявлення; не вимагає встановлення; є можливість резерв- ного копіювання даних.
CyberSafe	Є платною і вимагає спеціального встановлення.	Широкий спектр можливостей; необмежена довжина пароля для шифрування; використання алгоритмів шифрування; простота використання.
Folder Lock	Наявність тільки англійської мови; відсутня цифровий підпис; висока вартість.	Простий і зрозумілий інтерфейс; прозоре шифрування «на льоту».
Double Protection	Не весь діапазон форматів, який над яким можна виконувати операції.	Простота у використанні, додатковий захист шифруванням.

Таблиця 4.5 – Порівняльний аналіз переваг та недоліків в існуючих методах

Методи приховування інформації	Недоліки	Переваги
Методи, що використовують зарезервовані розширення полів.	Низька ступінь скритності, передача невеликих та обмежених обсягів інформації.	Простота використання.
Методи, що використовують відомі зміщенні слова, речення, абзаци.	Слабка продуктивність методу, можливість передачі невеликих обсягів інформації.	Простота використання.
Методи вибору певних позицій букв	Низька ступінь прихованості інформації.	Існує велика кількість вільно поширеного програмного забезпечення для реалізації цих методів.
Методи приховування інформації в деяких місцях гнучких дисків	Слабка продуктивність методу, можливість передачі невеликих обсягів інформації.	Простота використання.
Методи, що використовують надмірності файлів	Спотворюються статистичні характеристики цифрових потоків.	Можливість прихованої передачі великого обсягу інформації.
Розроблений метод	Не весь діапазон форматів, який над яким можна виконувати операції.	Великі затрати часу при взламі, які сприяють великій ймовірності втрати актуальності інформації.

4.3 Прогнозування витрат на виконання та впровадження результатів наукової роботи

У даній магістерській кваліфікаційній роботі розглядається програмне забезпечення для приховування текстової інформації у цифрових зображеннях, це по суті програма, а при розробці програми значна частина витрат - це витрати на розробку, а не на виробництво і відтворення. Звідси, й певна специфіка розрахунків.

Основна заробітна плата розробників, які працюють над проектом визначається за формулою 4.1:

$$Зо = \frac{М}{Тр} \cdot t, (грн.) \qquad (4.1)$$

де М - місячний посадовий оклад розробника;

Тр - число робочих днів в місяці (T_p = 21 день);

t - число днів роботи розробника.

Над створенням розробки працювали керівник проекту та інженер-програміст, отже, виконаємо для них всі необхідні розрахунки:

$$Зо = \frac{12000{,}00}{21} \cdot 4 = 2285{,}71 \text{ (грн)}.$$

$$Зо = \frac{16000{,}00}{21} \cdot 66 = 50285{,}71 \text{ (грн)}.$$

Таблиця 4.6 – Заробітна плата

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
1 Керівник	12000,00	571,42	4	2285,71
2 Інженер-програміст	16000,00	8337,66	66	50285,71
Всього				$\sum Зо = 52571{,}42$

Додаткова заробітна плата Зд всіх розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховується як (10…12)% від суми основної заробітної плати всіх розробників та робітників, тобто:

$$Зд = (10 \ldots 12\%) \cdot Зо, (грн). \qquad (4.2)$$

де Зо- основана заробітна плата.

$$Зд = \frac{10 \cdot 52571{,}42}{100} = 5257{,}14 (грн).$$

Нарахування на заробітну плату Нзп розробників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$Нзп = (22\%) \cdot (Зо + Зд), (грн). \qquad (4.3)$$

$$Нзп = \frac{22 \cdot (52571{,}42 + 5257{,}14)}{100} = 12722{,}28 \text{ (грн)}.$$

Амортизація обладнання, комп'ютерів та приміщень А, які використовувались під час (чи для) виконання даного етапу роботи.

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

У спрощеному вигляді амортизаційні відрахування А в цілому були розраховані за формулою 4.4:

$$А = \frac{Ц \cdot На}{100} \cdot \frac{Т}{12}, \text{(грн)}. \quad (4.4)$$

де Ц - загальна балансова вартість всього обладнання, комп'ютерів, приміщень тощо, що використовувались для виконання даного етапу роботи, грн;

На - річна норма амортизаційних відрахувань;

Т - термін, використання обладнання, приміщень тощо, місяці.

$$А_{ЕОМ} = \frac{14000 \cdot 20}{100} \cdot \frac{3}{12} = 700 \text{ (грн)}.$$

$$А_{ЕОМ} = \frac{9500 \cdot 20}{100} \cdot \frac{3}{12} = 475 \text{ (грн)}.$$

Всі розрахунки зводимо до таблиці 4.7.

Таблиця 4.7 - Амортизація обладнання та приміщень

Найменування обладнання, приміщень	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн.
ЕОМ	14000	20%	3	700,00
Приміщення для дипломного проектування	9500	20%	3	475,00
Всього				А = 1175,00

Для роботи над магістерською роботою були використані наступні матеріали.

Таблиця 4.8 – Послуги, що використовуються при виготовленні інноваційного продукту

Найменування комплектуючих (робіт, послуг)	Кількість, шт.	Ціна за одиницю, грн.	Сума, грн.
1. Аркуш	120	0,16	19,2
2. Флешка	1	120	120
3. Картридж для принтера	1	635,12	635,12
Всього	К = 774,32 грн.		

Витрати на силову електроенергію $В_е$, якщо ця стаття має суттєве значення для виконання даного етапу роботи, розраховуються за формулою 4.6:

$$В_е = В \cdot П \cdot Ф \cdot К_п, (грн). \quad (4.6)$$

де В - вартість 1 кВт електроенергії, грн.;

П - установлена потужність обладнання, кВт/год;

Ф - фактична кількість годин роботи обладнання, яке задіяне на виготовлення одного виробу, годин;

$К_п$ - коефіцієнт використання потужності, $К_п \leq 1$.

Вартість 1кВт електроенергії рівна 1,90 грн, потужність обладнання (ноутбука) рівна 90 Вт, що тотожно 0,09 кВт, фактична робота обладнання для роботи рівна 528 год, а коефіцієнт використання потужності приймемо за 0,8.

$$В_е = 1,90 \cdot 0,09 \cdot 528 \cdot 0,8 = 72,23 (грн).$$

Інші витрати. Інші витрати охоплюють: загально виробничі витрати (витрати управління організацією, ремонт та експлуатація основних засобів, витрати на опалення, освітлення тощо), адміністративні витрати (проведення

зборів, оплата юридичних та аудиторських послуг, тощо), витрати на збут (витрати на рекламу, перепідготовка кадрів) на інші операційні витрати (штрафи, пені, матеріальні допомоги, втрати від знецінення запасів тощо).

Інші витрати можна розрахувати за нормативами відносно основної заробітної плати основних робітників, які виготовляють продукцію, за формулою 4.7.

$$В_{ін} = Н \cdot Зо, (грн). \qquad (4.7)$$

де Н - норматив загальновиробничих витрат. Для ЕОМ Н = 100%.

$$В_{ін} = \frac{100 \cdot 52571{,}42}{100} = 52571{,}42 \ (грн).$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини (розділу, етапу) роботи - В.

$$В = 52571{,}42 + 5257{,}14 + 12722{,}28 + 1175{,}00 + 774{,}32 + 72{,}23 + 52571{,}42 = 125143{,}81 \ (грн).$$

Прогнозування загальних витрат ЗВ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{В_{заг}}{\beta}, \qquad (4.8)$$

де β - коефіцієнт, який характеризує етап (стадію) виконання даної роботи. Оскільки, проект знаходиться на стадії розробки, то $\beta \approx 0{,}7$;

$В_{заг}$ - загальна вартість всієї наукової роботи. У даному випадку $В_{заг} = В$.

$$ЗВ = \frac{125143{,}81}{0{,}7} = 178776{,}87 (грн).$$

Отже, розрахований кошторис витрат на розробку програмного забезпечення для приховування текстової інформації у цифрових зображеннях складає 178776,87 грн.

4.4 Прогнозування комерційних ефектів від реалізації результатів розробки

У даному підрозділі виконано прогнозування, яку вигоду можна отримати у майбутньому від впровадження результатів даної наукової роботи.

Передбачається, що виконання наукової роботи та впровадження результатів по розробці програмного забезпечення для приховування текстової інформації у цифрових зображеннях зображень займе 1 рік.

Основні позитивні результати від впровадження розробки очікуються протягом 3 років після її впровадження.

Саме зростання чистого прибутку забезпечить підприємству (організації) надходження додаткових коштів, які дозволять покращити фінансові результати діяльності.

$$\Delta П_i = \sum_{1}^{n} (\Delta П_я \cdot N + П_я \Delta N)_i, (грн). \quad (4.9)$$

де $\Delta П_я$ – покращення основного якісного показника від впровадження результатів розробки у даному році;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

$П_я$ – основний якісний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

В результаті впровадження результатів наукової розробки покращується якість програмного продукту, що дозволяє підвищити ціну його реалізації на 270 грн., а кількість потенційних користувачів ресурсу

збільшиться: протягом першого року - на 800 шт., протягом другого року - ще на 1600 шт., протягом третього року - ще на 20000 шт.

Орієнтовно: реалізація продукції до впровадження результатів наукової розробки складала 1шт., а прибуток, що його отримувало підприємство на одиницю продукції до впровадження результатів наукової розробки – 60 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового.

Збільшення чистого прибутку підприємства протягом наступних трьох років складе:

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом першого року складе:

$$\Delta П_1 = 60 \cdot 1 + (60 + 100) \cdot 800 = 128060 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta П_i$ протягом другого року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П_2 = 60 \cdot 1 + (60 + 100) \cdot 1600 = 256060 \text{ (грн)}.$$

Збільшення чистого прибутку підприємства $\Delta П_i$протягом третього року (відносно базового року, тобто року до впровадження результатів наукової розробки) складе:

$$\Delta П_3 = 60 \cdot 1 + (60 + 100) \cdot 20000 = 3200060 \text{ (грн)}.$$

4.5 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахований комерційний ефект від можливого впровадження розробки ще не означає, що ця розробка реально буде впроваджена. Якщо збільшення прогнозованого прибутку від впровадження результатів наукової розробки є вигідним для підприємства, то це ще не означає, що інвестор погодиться фінансувати розробку.

Основними показниками, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахунок ефективності вкладених інвестицій передбачає проведення таких робіт:

1-й крок. Розрахуємо теперішню вартість інвестицій PV, що вкладаються в наукову розробку. Такою вартістю, можна вважати прогнозовану величину загальних витрат ЗВ на виконання та впровадження результатів НДДКР, розраховану нами раніше за формулою (4.8), тобто будемо вважати, що ЗВ = PV=178776,87.

2-й крок. Розрахуємо очікуване збільшення прибутку $\Delta П_i$, що його отримає підприємство (організація) від впровадження результатів наукової розробки, для кожного із років, починаючи з першого року впровадження. Таке збільшення прибутку, також було розраховане раніше за формулою 4.9.

3-й крок. Для спрощення подальших розрахунків побудуємо вісь часу, на яку нанесемо всі платежі (інвестиції та прибутки), що мають місце під час виконання науково-дослідної роботи та впровадження її результатів.

Платежі показуються у ті терміни, коли вони здійснюються. Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, наведений на рис. 4.2.

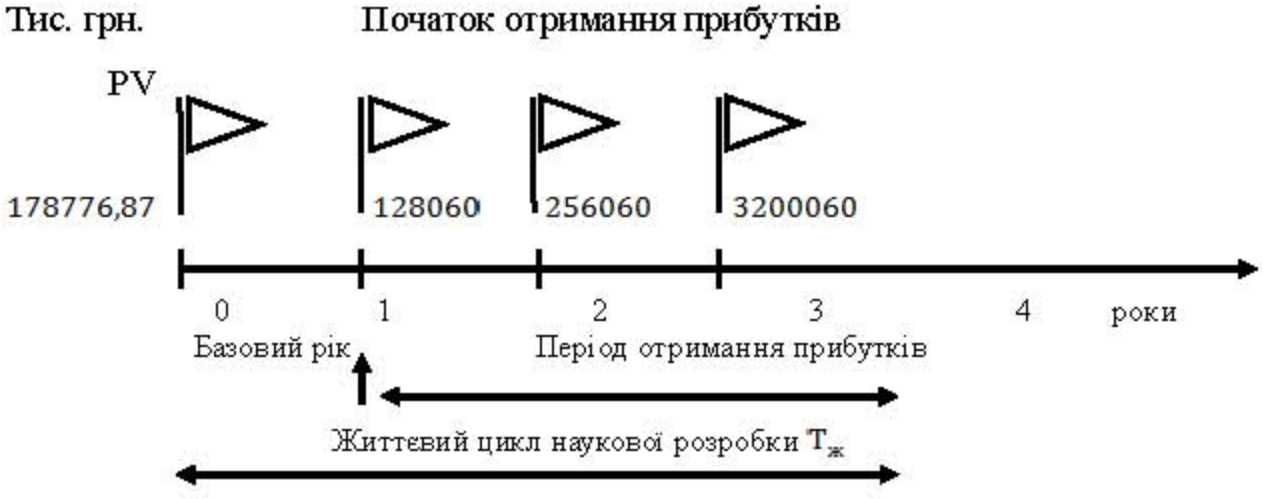

Рисунок 4.2 - Вісь часу з фіксацією платежів, що мають місце під час

4-й крок. Розрахуємо абсолютну ефективність вкладених інвестицій $Е_{абс}$.

Для цього користуються формулою:

$$Е_{абс} = (ПП - PV), \qquad (4.10)$$

де ПП - приведена вартість всіх чистих прибутків, що їх отримає підприємство (організація) від реалізації результатів наукової розробки, грн;

PV - теперішня вартість інвестицій PV = ЗВ = 178776,87 грн.

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, (тис. грн.) \qquad (4.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

Т – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки «0».

Отримаємо:

$$ПП = \frac{128060}{(1+0,1)^2} + \frac{256060}{(1+0,1)^3} + \frac{3200060}{(1+0,1)^4} = \frac{128060}{1,21} + \frac{256060}{1,33} + \frac{3200060}{1,46} = \frac{128060 \cdot 1,93 + 256060 \cdot 1,75 + 3200060 \cdot 1,60}{2,34} = \frac{5815356,8}{2,34} = 2485195,21$$

тис. грн.

Тоді $Е_{абс} = 2485195,21 - 178776,87 = 2306418,34$ тис. грн.

Оскільки $E_{абс} > 0$, то результат від проведення наукових досліджень та їх впровадження може принести прибуток, але це також ще не свідчить про те, що інвестор буде зацікавлений у фінансуванні даної роботи.

5-й крок. Розраховують відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього користуються формулою:

$$E_в = \sqrt[Т_ж]{1 + \frac{E_{абс}}{PV}} - 1, (\%) \qquad (4.12)$$

де $E_{абс}$ - абсолютна ефективність вкладених інвестицій, грн;

PV - теперішня вартість інвестицій PV = ЗВ, грн;

$Т_ж$ - життєвий цикл наукової розробки, роки.

$$E_в = \sqrt[4]{1 + \frac{2306418,34}{178776,87}} - 1 = \sqrt[4]{13,90} - 1 = 0,93 = 93\ \%$$

Далі, розрахована величина $E_в$ порівнюється з мінімальною (бар'єрною) ставкою дисконтування $\tau_{мін}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{мін}$ визначається за формулою:

$$t = d + f, (\%) \qquad (4.13)$$

де d – середньо зважена ставка за депозитними операціями в комерційних банках; в 2019 році в Україні d = (0,14...0,2);

f - показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,1), але може бути і значно більше.

$$t = d + f = 0,2 + 0,1 = 0,3 = 30\%$$

Величина $E_в > \tau_{мін}$, інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених коштів у реалізацію наукового

проекту за формулою:

$$T_{ок} = \frac{1}{E_в}, (років) \quad (4.14)$$

$$T_{ок} = \frac{1}{0,93} = 1,07 \text{ року.}$$

Оскільки $T_{ок}$ = 1,07 року (один рік), то фінансування розробки є доцільним.

4.6 Висновок до розділу

У четвертому розділі магістерської кваліфікаційної роботи проведено розрахунки, що доводять економічну доцільність та ефективність впровадження розробленого продукту. Розрахунки поділено на 4 частини, які утворили відповідні підрозділи цього розділу. У комплексі ці підрозділи дозволяють побачити цілісну картину економічної доцільності нового інтелектуального рішення.

У першому підрозділі здійснено оцінювання комерційного потенціалу розробки. На основі компетентної думки експертів було сформовано систему критеріїв та за 5-ти бальною шкалою, виставлено відповідні бали по кожному критерієві. Виставлені бали, говорять про те, що рівень комерційного потенціалу є середнім.

Другий підрозділ економічної частини демонструє витрати на розробку, що розраховуються, як сума усіх статей витрат поділена на ступінь готовності продукту. Розрахований кошторис витрат на розробку складав 178776,87 грн.

Далі прогнозується комерційний ефект від реалізації розробки, тобто яку вигоду, можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

Останній підрозділ висвітлює основі показники, які визначають доцільність фінансування наукової розробки певним інвестором. Такими показниками є абсолютна та відносна ефективність вкладених інвестицій, а

також термін їх окупності.

Обрахована абсолютна ефективність становить 2306418,34 тис. грн, що свідчить про те, що інвестор буде зацікавлений у фінансуванні даної розробки.

Відносна (щорічна) ефективність становить 93%, що більше мінімальної ставки дисконтування, що ще раз підтверджує зацікавленість інвестора.

Термін окупності вкладених коштів у реалізацію наукового проекту становить 1,07 що означає, що вкладені кошти повернуться через один рік.

Таким чином, можна стверджувати, що фінансування даної розробки є доцільним.

ВИСНОВКИ

У даній магістерській дисертації було розроблено покращений алгоритм, який заснований на приховуванні за допомогою найменш значущого біта. Модифікований він тим, що порядок вибору найменш значущого біта є довільним, а це в свою чергу дає можливість уникнути вияв факту передачі за допомогою методу Хі квадрат та RS-атак. Стеганографічний алгоритм приховування працює у поєднанні з криптографічним алгоритмом RSA, через його можливі варіації та стійкість до зламу. Така комбінація дозволяє не тільки приховувати вміст повідомлення, для передачі по відкритому або не захищеному каналу зв'язку, а й додатково створити захист на випадок якщо повідомлення все ж таки буде виявлено.

В результаті першого етапу розробки розглянуто основні етапи розвитку направлення стеганографії, проведено аналіз основних понять, компонентів і методів комп'ютерної стеганографії, які пов'язані з вбудовуванням тексту в зображення різних форматів. Для вирішення поставленої задачі створено поетапний та структурований перелік завдань.

В другому розділі проведено аналіз та порівняння програмних засобів, розглянуто методики та алгоритми приховування інформації, запропоновано модифікації алгоритму приховування. Розглянуто метод виявлення вкраплень за допомогою RS-атак. Також обґрунтовано вибір стеганографічного алгоритму над яким проведено покращення та описаний алгоритм шифрування заповненого контейнеру. За допомогою даної комбінації алгоритмів відбувається уникнення вияву прихованого повідомлення методом Хі-квадрат і RS-атак та надається додатковий захист на випадок, якщо зловмиснику все ж вдалось виявити факт передачі.

У межах третього розділу описано архітектуру програмного продукту, які засоби були використані для створення веб-додатку, з яких кроків складається робота програми. Створена інструкція користувачу, описані частини коду для роботи програми, приведена робота веб-додатку у графічному вигляді. Під час тестування збоїв не було. Обрана комбінація

методів передбачає безпечніше приховування тексту, стійкість до вияву факту вбудовування, додатковий захист у вигляді шифрування. Інтерфейс є максимально простим та інтуїтивно зрозумілим.

У четвертому розділі були проведені розрахунки по економічній доцільності розробки нового програмного продукту по приховуванню текстової інформації у зображеннях. Проведено розрахунки комерційного потенціалу, визначені витрати на розробку, виконано прогнозування комерційного ефекту, визначено термін окупності, обрахована абсолютна та відносна ефективність. Комплексно ці розрахунки дозволили створити цілісну картину економічної доцільності нового інтелектуального рішення.

Розроблений алгоритм є більш стійким до вияву і має в наявності захист у вигляді шифрування. Веб-додатку необхідні допрацювання у вигляді розширення форматів зображень з якими передбачається робота та налагодження роботи серверної частини програми, але в цілому перелік поставлених задач виконаний.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Грибунин В.Г. Цифровая стеганография / В.Г. Грибунин, И.Н. Оков, И.В. Туринцев – М.: СОЛОН-Пресс, 2012 – 274 с.

2) Б. Шнайер. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. – М.: Издательство ТРИУМФ, 2014 – 816с.: ил.

3) Fabien A. P. Petitcolas, Ross J. Anderson and Markus G. uhn. Information Hiding: A Survey // Proceedings of the IEEE (special issue). — 1999. — Vol. 87, no. 7. — P. 1062–1078. — DOI:10.1109/5.771065.- https://www.petitcolas.net/fabien/publications/ieee99-infohiding.pdf

4) Reeds, Jim (1998). Solved: The ciphers in book III of Trithemius's Steganographia. *Cryptologia*. Архів оригіналу за 29 січень 2005. Процитовано 1 червень 2013. (PDF, 209KiB) - https://web.archive.org/web/20050129212947/http://www.dtc.umn.edu/~reedsj/trit.pdf

5) 2001 Intel Science Talent Search Winners. - https://www.infoplease.com/arts-entertainment/intel-science-talent-search-winners/2001-intel-science-talent-search-winners

6) Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. – К.: МК-Пресс, 2006. – 288 с.

7) Серов Р.Е., Гончаров В.В., Основы современной криптографии – Москва, Горячая линия – Телеком, 2011. – 443 с.

8) Shapiro J. Embedded Image Coding Using Zerotrees Of Wavelet Coefficients / / IEEE Transactions on Signal Processing, 1993. – Vol. 41, No. 12.

9) Быков С.Ф., Мотуз О.В. Основы стегоанализа.// Защита информации. Конфидент. – СПб.: 2000, № 3. – С. 38-41.

10) Грибунин В.Г. Цифровая стеганография / В.Г.Грибунин, И.Н.Оков, И.В.Туринцев. – М. : СОЛОН-Пресс, 2002. – 272 с

11) Клопов В.А. Основы компьютерной стеганографии / В.А. Клопов, О.В. Мотуз // Конфидент. – 1997. – №4. – С.43–48.

12) Кустов В.Н. Методы встраивания скрытых сообщений / В.Н. Кустов, А.А. Федчук // Конфидент. – 2000. – № 3. – С.34–37.

13) Задірака В.К. Спектральні алгоритми комп'ютерної стеганографії / В.К. Задірака, С.С. Мельнікова, Н.В. Бородавка //Искусственный интеллект. – 2002. – № 3. – С. 532 –541.

14) Швидченко И.В. Методы стеганоанализа для графических файлов / И.В. Швидченко // Искусственный интеллект. – 2010. – № 4. – С. 697–705.

15) Швідченко І.В. Аналіз програмного забезпечення зі стеганоаналізу / І.В. Швідченко //Искусственный интеллект. – 2012. – №3. – С. 487–495.

16) Suresh A. Image Texture Classification using Gray Level Co-Occurrence Matrix Based Statistical Features / A. Suresh, K.L. Shunmuganathan // European Journal of Scientific Research. – 2012. – Vol.75, № 4. – P. 591–597

17) Певнев В.Я. RSA и простые числа // Системи обробки інформації, 2016, випуск 8 (145).- С.118-120.

18) Методичні вказівки до виконання студентами-магістрантами економічної частини магістерських кваліфікаційних робітhttp://lglushenko.vk.vntu.edu.ua/file/3bc877773dd4cff5ecd0c5fca5e7c5f2.pdf